# Articulated Human Tracking and Behavioural Analysis in Video Sequences

**Zsolt Levente Husz**

Submitted for the degree of
Doctor of Philosophy

Heriot-Watt University
School of Engineering and Physical Sciences

October, 2008

# Abstract

Recently, there has been a dramatic growth of interest in the observation and tracking of human subjects through video sequences. Arguably, the principal impetus has come from the perceived demand for technological surveillance, however applications in entertainment, intelligent domiciles and medicine are also increasing. This thesis examines human articulated tracking and the classification of human movement, first separately and then as a sequential process.

First, this thesis considers the development and training of a 3D model of human body structure and dynamics. To process video sequences, an observation model is also designed with a multi-component likelihood based on edge, silhouette and colour. This is defined on the articulated limbs, and visible from a single or multiple cameras, each of which may be calibrated from that sequence. Second, for behavioural analysis, we develop a methodology in which actions and activities are described by semantic labels generated from a *Movement Cluster Model* (MCM). Third, a *Hierarchical Partitioned Particle Filter* (HPPF) was developed for human tracking that allows multi-level parameter search consistent with the body structure. This tracker relies on the articulated motion prediction provided by the MCM at pose or limb level. Fourth, tracking and movement analysis are integrated to generate a probabilistic activity description with action labels.

The implemented algorithms for tracking and behavioural analysis are tested extensively and independently against ground truth on human tracking and surveillance datasets. Dynamic models are shown to predict and generate synthetic motion, while MCM recovers both periodic and non-periodic activities, defined either on the whole body or at the limb level. Tracking results are comparable with the state of the art, however the integrated behaviour analysis adds to the value of the approach.

# Acknowledgements

First, I would like to say thanks to all whom during the last three years have helped accomplishing this thesis by ideas and suggestions, daily online messages, friendly chats, long-distance phone calls, or with warm handshakes.

Particularly, I thanks my supervisors, Prof. Andrew M. Wallace and Dr. Patrick R. Green, for the professional support that I have received. I acknowledge the support of Heriot-Watt University and ORS AS.

Especially, I say thanks to my supporting family.

# Table of Contents

**List of Tables** **ix**

**List of Figures** **xi**

**List of Abbreviations and Symbols** **xv**

**1 Introduction** **1**

1.1 Motivation . . . . . 1

1.1.1 Public interest . . . . . 1

1.1.2 Scientific motivation . . . . . 2

1.2 Project overview . . . . . 3

1.3 The contributions of the thesis . . . . . 4

1.3.1 Prior information modelling . . . . . 4

1.3.2 Articulated tracking . . . . . 4

1.3.3 Action recognition and behavioural analysis . . . . . 4

1.3.4 Unified tracking-analysis framework . . . . . 5

1.4 Thesis outline . . . . . 5

**2 A survey of behaviour and motion recovery** **6**

2.1 Understanding human behaviour . . . . . 6

2.1.1 Psychological evidence . . . . . 7

2.1.2 Behavioural systems: services and requirements . . . . . 9

2.1.3 Psychologically inspired system models . . . . . 10

2.1.4 Pose, movement, action, activity, behaviour and gesture . . . . . 12

2.1.5 Behaviour understanding and human model recovery . . . . . 16

2.1.6 Conclusions . . . . . 22

2.2 Methodologies for pose recovery and tracking . . . . . 22

2.2.1 Deterministic tracking . . . . . 23

2.2.2 Stochastic tracking . . . 29
2.2.3 Conclusions . . . 37
2.3 The use of prior knowledge in tracking . . . 38
2.3.1 Human body models . . . 39
2.3.2 Human dynamics . . . 46
2.3.3 Scene models . . . 49
2.3.4 Behavioural priors . . . 53
2.3.5 Conclusions . . . 54
2.4 Image measurements and likelihood functions . . . 55
2.4.1 Silhouette . . . 57
2.4.2 Edge . . . 57
2.4.3 Colour . . . 58
2.4.4 Texture . . . 59
2.4.5 Multiple camera observation . . . 59
2.4.6 Non-visual observation . . . 60
2.4.7 Measurement fusion . . . 61
2.5 Human tracking systems . . . 62
2.6 Training and evaluation data . . . 65
2.6.1 HumanEva dataset . . . 65
2.6.2 CAVIAR dataset . . . 66
2.6.3 i-LIDS . . . 67
2.6.4 Other tracking datasets . . . 68
2.6.5 Other behavioural datasets . . . 69
2.7 Performance evaluation . . . 69
2.8 Summary . . . 72

**3 Models for human tracking** **75**

3.1 The three-dimensional space and the camera model . . . 75
3.1.1 Calibration with vanishing points . . . 79
3.1.2 Calibration with point correspondences . . . 83
3.1.3 Conclusions on calibration . . . 85
3.1.4 Test sequence calibration . . . 85
3.2 The Articulated Hierarchical Human Model . . . 88

3.2.1 Parametrisation of the AHHM . . . 90
3.2.2 Parameter range constraint . . . 91
3.2.3 The limb coordinate systems . . . 91
3.2.4 Body part projections . . . 94
3.2.5 The self-occlusion reasoning . . . 95
3.2.6 The Maximum Visibility prior . . . 96
3.2.7 Three-dimensional humanoid structure test . . . 96
3.2.8 Comparison with HumanEva model . . . 98
3.3 Observation model . . . 100
3.3.1 Likelihood composition . . . 101
3.3.2 Silhouette based likelihoods . . . 103
3.3.3 Edge based likelihood . . . 104
3.3.4 Colour likelihood . . . 105
3.3.5 Global likelihood . . . 106
3.4 Summary . . . 106

**4 Human dynamics and behaviour modelling 108**
4.1 Body feature vector and movement clusters . . . 109
4.2 Pose Transitional Model . . . 112
4.2.1 Pose compression . . . 114
4.2.2 Cluster formation . . . 115
4.2.3 Cluster modelling . . . 117
4.2.4 Transition probabilities . . . 117
4.2.5 Algorithm output . . . 117
4.2.6 Synthetic motion generation . . . 118
4.3 Continuous transition models . . . 121
4.4 Continuous Time Pose Transition Model . . . 121
4.4.1 Synthetic motion generation . . . 123
4.5 Movement Cluster Model . . . 125
4.5.1 Compression of movements . . . 126
4.5.2 Motion generation . . . 127
4.5.3 Model learning . . . 129
4.5.4 Movement likelihood and movement conditioned MC probability . . 133

4.5.5 Experiment: MC uniformity . . . 134
4.5.6 Synthetic motion generation . . . 137
4.6 Behaviour primitives . . . 140
4.6.1 Action learning . . . 142
4.6.2 Action labels of the HumanEva dataset . . . 142
4.6.3 MCM behavioural analysis of the HumanEva dataset . . . 144
4.6.4 Actions from the MCM sets . . . 146
4.6.5 Recognition evaluation . . . 148
4.6.6 Recognition sensitivity . . . 151
4.7 Activity reasoning . . . 152
4.8 Summary . . . 155

**5 Articulated human tracking** **158**
5.1 Evaluation methodology . . . 159
5.2 Stochastic tracking . . . 160
5.2.1 Uniform and normal distributions. Sampling . . . 160
5.2.2 Kalman filters . . . 161
5.2.3 Particle Filter basics and Sequential Importance Resampling . . . 163
5.2.4 The Partitioned Particle Filter . . . 169
5.2.5 The Annealed Particle Filter . . . 169
5.3 Hierarchical tracking . . . 172
5.3.1 Architecture . . . 173
5.3.2 Algorithmic description . . . 175
5.3.3 HPPF for AHHM . . . 177
5.3.4 Quantitative evaluation of PFs for AHHM . . . 177
5.3.5 HPPF and other PFs . . . 180
5.3.6 HPPF with MCM . . . 180
5.3.7 Model complexity . . . 184
5.4 Parameter adjustment for HPPF with MCM . . . 184
5.4.1 Motion model parameters . . . 186
5.4.2 Number of particles . . . 186
5.4.3 The tracking estimate . . . 190
5.4.4 Likelihoods and priors . . . 191

5.4.5 Stochastic constants . . . 193
5.4.6 Particle survival . . . 193
5.4.7 Optimised parameters . . . 195
5.5 Multiple vs. single camera tracking . . . 195
5.6 Tracking results . . . 196
5.6.1 HumanEva-I test sequences . . . 197
5.6.2 HumanEva-II test sequences . . . 197
5.6.3 Tracking CAVIAR sequences . . . 214
5.6.4 iLIDS sequences . . . 214
5.7 Summary and conclusions . . . 224

**6 Combining tracking and behavioural analysis 227**
6.1 Behaviour from the tracked model . . . 227
6.2 The influence of the MCM parameters . . . 229
6.3 Tracking and analysis with independent models . . . 232
6.4 The influence of the MCM detail . . . 232
6.5 Recognition of HumanEva sequences . . . 233
6.6 Recognition with reduced camera number . . . 238
6.7 Recognition of the CAVIAR sequence . . . 238
6.8 Discussion . . . 238

**7 Conclusions and future work 244**
7.1 Summary and contributions . . . 244
7.2 Future work . . . 246

**A Publications 250**

**Bibliography 251**

# List of Tables


2.1 Behavioural systems intelligent services. . . . . 10
2.2 Behavioural systems requirements. . . . . 10
2.3 Pose, movement, gesture, action and activity. . . . . 14
2.4 Stochastic filter comparison chart. . . . . 31
2.5 PF based human tracking. . . . . 34
2.6 Advantages and disadvantages of PFs. . . . . 36
2.7 Survey of the current algorithm priors, measurements and methodology. . . 40
2.8 Three-dimensional human models. . . . . 44
2.9 Likelihood for generative approaches. . . . . 56

3.1 Recovered principal point and focal length I. . . . . 81
3.2 Recovered principal point and focal length II. . . . . 82
3.3 Recovered principal point and focal length III. . . . . 82
3.4 Calibration and test points for the CAVIAR corridor scene. . . . . 87
3.5 Calibration and test points for the CAVIAR frontal scene. . . . . 87
3.6 Calibration and test points for the i-LIDS scene. . . . . 88
3.7 Recovered calibration parameters. . . . . 89
3.8 Limb coordinate systems definitions. . . . . 91
3.9 The 24 parameters of the pose vector. . . . . 92
3.10 Parametrisation of body parts. . . . . 93
3.11 Similarities and differences of the AHHM and HumanEva models. . . . . 98
3.12 HumanEva to AHHM transformations by rotations . . . . . 99

4.1 PCA compression of poses. . . . . 116
4.2 PCA compression of movements. . . . . 127
4.3 The set of MCMs. . . . . 132
4.4 Cluster uniformity. . . . . 136

4.5 Local labels with descriptions and training sequences. . . . . 143
4.6 Accuracy variation for added noise. . . . . 153

5.1 Test sequences for articulated human tracking. . . . . 159
5.2 Partition definition per each level. . . . . 178
5.3 Particle filter variant accuracy. . . . . 179
5.4 Particle filter variant speed. . . . . 179
5.5 Propagation type definition. . . . . 181
5.6 Errors of HPPF for MCM propagation modes. . . . . 183
5.7 Errors for simplified model human model. . . . . 184
5.8 Tracking error dependence on $n_{\mathcal{C}}$ and $l_m$. . . . . 187
5.9 Tracking error dependence on the number of particles. . . . . 188
5.10 Errors statistics on the number of particles. . . . . 189
5.11 Tracking error dependence on the estimation method. . . . . 191
5.12 Tracking error dependence on the likelihood function on supplementary priors. 192
5.13 Tracking error dependence on propagation stochastic constants. . . . . 194
5.14 Tracking result for fitness survival. . . . . 195
5.15 Optimised parameters of the HPPF-MCM tracker. . . . . 196
5.16 Tracking error for reduced camera input. . . . . 197
5.17 Tracking errors for seven HumanEva sequences. . . . . 197
5.18 Absolute 3D and 2D tracking errors for the three HumanEva 2 test sequences. 198
5.19 Tracking errors compared with state of the art. . . . . 210
5.20 Comparing HPPF with state of the art. . . . . 212
5.21 Summary of tracking effects of the HPPF-MCM parameters . . . . . 225

6.1 Recognition accuracy with identical $\mathcal{M}_1$ MCMs for both tracking and behavioural analysis. . . . . 230
6.2 Recognition accuracy with independent $\mathcal{M}_1$ MCMs for both tracking and behavioural analysis. . . . . 232
6.3 Recognition accuracy with $\mathcal{M}_7$ MCMs. . . . . 233

# List of Figures


1.1 Separation of tracking and behavioural analysis. . . . . 3

2.1 Detecting irregularities with space & time correlation. . . . . 18
2.2 Codebook based pedestrian detection. . . . . 24
2.3 Partitioned particle filter. . . . . 33
2.4 APF convergence over multiple layers. . . . . 35
2.5 Nonparametric belief propagation. . . . . 38
2.6 Instances of the SCAPE model. . . . . 43
2.7 Motion prediction and generation with database lookup. . . . . 49
2.8 A scene model example . . . . . 50
2.9 Reading tracker run on a CAVIAR sequence. . . . . 64
2.10 Example frames of the HumanEva dataset. . . . . 65
2.11 CAVIAR dataset examples. . . . . 66
2.12 i-LIDS Abandoned baggage scenario . . . . . 67

3.1 The relations between world, camera and image coordinate systems.. . . . . 76
3.2 Definition of vanishing points. . . . . 79
3.3 Test cube with high and low perspective transformation. . . . . 81
3.4 Calibration tool of an uncalibrated image. . . . . 84
3.5 Calibration test with the Reading Tracker. . . . . 85
3.6 CAVIAR corridor view calibration. . . . . 86
3.7 CAVIAR frontal view calibration. . . . . 87
3.8 i-LIDS calibration. . . . . 88
3.9 A body pose with the AHHM. . . . . 90
3.10 Neutral configuration of the LCSs. . . . . 93
3.11 The projected visible edges. . . . . 95
3.12 Depth map example. . . . . 96

3.13 Left leg poses . . . . . 97
3.14 HumanEva dataset artefacts. . . . . 100
3.15 Multiple measurements. . . . . 103

4.1 Movement clusters and actions. . . . . 110
4.2 BFVs, movements, actions and activities. . . . . 111
4.3 Distribution of actions for $n_{\mathcal{C}} = 60$ MCs. . . . . 111
4.4 Motion model learning overview. . . . . 112
4.5 First 20 poses of random motion generated with the PTM [◊]. . . . . 119
4.6 A transition sequence with PTM. . . . . 120
4.7 Transition duration definition . . . . . 122
4.8 Continuous Time Pose Transition Model. . . . . 123
4.9 Random motion generated with the CTPTM [◊]. . . . . 124
4.10 A transition sequence with CTPTM. . . . . 125
4.11 Visual example of a MCM. . . . . 126
4.12 Block diagram of the MCM learning algorithm. . . . . 131
4.13 Movement length dependent cluster composition. . . . . 134
4.14 MC number dependent dependent cluster composition. . . . . 135
4.15 Left leg random motion with MCM [◊]. . . . . 138
4.16 A leg MC transition sequence. . . . . 138
4.17 Full body random motion with MCM [◊]. . . . . 139
4.18 A whole-pose MC transition sequence. . . . . 139
4.19 Whole-pose MC transition sequences for 2000 MCs. . . . . 141
4.20 The labelling interface. . . . . 145
4.21 Recognition with known and unknown human subject. . . . . 146
4.22 *S1 Walking 1* activity recognition. . . . . 147
4.23 Confusion matrices for MCM comparison. . . . . 148
4.24 Recognition for the MCMs set on the *S1 Walking 1* sequence. . . . . 149
4.25 Recognition for the MCMs set on the *S3 Walking 1* sequence. . . . . 150
4.26 The absolute and relative errors for noisy parameters. . . . . 151
4.27 Action recognition with added noise. . . . . 152
4.28 Accuracy variation for added noise . . . . . 154
4.29 Confusion matrices of sequence classification. . . . . 155

5.1 PF-SIR tracking: 2D poses [◊]. . . . . 166
5.2 PF-SIR tracking: 3D poses [◊]. . . . . 167
5.3 PF-SIR tracking: *S1 Walking 1* 3D error. . . . . 168
5.4 PF-SIR tracking with elimination: *S1 Walking 1* 3D error. . . . . 168
5.5 APF tracking: *S1 Walking 1* 2D poses [◊]. . . . . 170
5.6 APF tracking: *S1 Walking 1* 3D reconstruction [◊]. . . . . 171
5.7 APF tracking: *S1 Walking 1* 3D error. . . . . 172
5.8 APF variance evolution. . . . . 173
5.9 HPPF particle set evolution. . . . . 174
5.10 HPPF for human tracking. . . . . 178
5.11 Reduced complexity human model tracked with the HPPF [◊]. . . . . 185
5.12 The effect of particle number on the 3D absolute error. . . . . 188
5.13 Particle number statistical analysis. . . . . 189
5.14 Tracking estimate. . . . . 190
5.15 HumanEva *S1 Walking 1* camera C2 sequence [◊]. . . . . 199
5.16 HumanEva *S1 Walking 1* 3D reconstruction [◊]. . . . . 200
5.17 HumanEva *S2 Combo 1* camera C1 sequence [◊]. . . . . 202
5.18 HumanEva *S2 Combo 1* camera C2 sequence [◊]. . . . . 203
5.19 HumanEva *S2 Combo 1* camera C3 sequence [◊]. . . . . 204
5.20 HumanEva *S2 Combo 1* camera C4 sequence [◊]. . . . . 205
5.21 HumanEva *S2 Combo 1* 3D reconstruction [◊]. . . . . 206
5.22 HumanEva *S4 Combo 4* camera C2 sequence I [◊]. . . . . 207
5.23 HumanEva *S4 Combo 4* camera C2 sequence II [◊]. . . . . 208
5.24 HumanEvaII 2D and 3D errors per frame. . . . . 209
5.25 Tracking comparison on the *S1 Walking 1* sequence. . . . . 211
5.26 Tracking comparison on the *S2 Combo 1* sequence. . . . . 212
5.27 Tracking comparison on the *S4 Combo 4* sequence. . . . . 213
5.28 CAVIAR EnterExitCrossingPaths1 corridor sequence [◊]. . . . . 215
5.29 CAVIAR EnterExitCrossingPaths1 frontal sequence [◊]. . . . . 216
5.30 CAVIAR EnterExitCrossingPaths1 3D reconstruction [◊]. . . . . 217
5.31 CAVIAR OneLeaveShopReenter1 corridor sequence [◊]. . . . . 218
5.32 CAVIAR OneLeaveShopReenter1 corridor sequence II [◊]. . . . . 219
5.33 CAVIAR OneLeaveShopReenter1 3D reconstruction I [◊]. . . . . 220

5.34 CAVIAR OneLeaveShopReenter1 3D reconstruction II [◊]. . . . . 221
5.35 i-LIDS AVSS AB Easy sequence 2D view [◊]. . . . . 222
5.36 i-LIDS AVSS AB Easy sequence 3D reconstruction [◊]. . . . . 223

6.1 Tracking and behavioural subsystem integration. . . . . 228
6.2 Confusion matrix dependence on MCM parameters. . . . . 231
6.3 Accuracies for full body and left upper arm partitions. . . . . 233
6.4 HumanEva *S1 Walking 1* sequence recognition. . . . . 235
6.5 The recovered HumanEva *S1 Walking 1* labels superimposed with the input frames [◊]. . . . . 236
6.6 HumanEva *S1 Gesture 1* sequence recognition. . . . . 237
6.7 HumanEva *S1 Jog 1* sequence recognition. . . . . 239
6.8 Recognition rate with reduced camera number. . . . . 239
6.9 CAVIAR *EnterExitCrossingPaths1* sequence recognition. . . . . 240
6.10 The recovered CAVIAR *EnterExitCrossingPaths1* labels superimposed with the input frames [◊]. . . . . 241

For figures marked with [◊], the full video sequence is available on the CD-ROM included with the thesis.

# List of Abbreviations and Symbols

### General conventions

- matrices and vector are straight letters (*e.g.* A and x);
- scalars are in italic letters (*e.g.* $s$);
- probability is noted with $\mathcal{P}$;
- standard operators and functions are in lower case and straight (*e.g.* chamf, cos);
- custom functions are alphanumeric strings in typewriter face (*e.g.* `F`);
- covariance matrices are engrossed (*e.g.* $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$);
- ${}_k\mathrm{x} = \mathrm{a}_k$ denotes the $k$-th sub-vector of x where $\mathrm{x} = [\mathrm{a}_n \ \mathrm{a}_{n-1} \ \ldots \ \mathrm{a}_0]$ has n+1 sub-vectors $\mathrm{a}_i$ with equal length;
- $\mathrm{x}^\phi = [x^{\phi_1}, \ldots, x^{\phi_m}]$ is a sub-vector of the vector $\mathrm{x} = [x^1, \ldots, x^l]$ with length $l$ for the partition and $\phi = \{\phi_1, \ldots, \phi_m\} \subset \{1, \ldots, l\}$;
- $\mathrm{x}_t$ is the value x at time $t$;
- ${}^m\mathrm{x}$ is value of x on the $m$-th level or iteration.

## List of Abbreviations

| | |
|---|---|
| 0GM | zero-order Gaussian Motion |
| 1GM | first-order Gaussian Motion |
| AHHM | Articulated Hierarchical Human Model |
| APF | Annealed Particle Filter |
| BFV | Body Feature Vector |
| CCS | camera coordinate system |
| CTPTM | Continuous Time Pose Transition Model |
| DOF | Degree of Freedom |
| EKF | Extended Kalman Filter |
| EKPF | Extended Kalman Particle Filter |
| EM | Expectation Maximisation |
| fps | frames per second |
| GPC | Ground Plane Constraint |
| HMM | Hidden Markov Model |
| HPPF | Hierarchical Partitioned Particle Filter |
| ICS | Image Coordinate System |
| KF | Kalman Filter |
| KLT | Kanade-Lucas-Tomasi (feature tracker) |
| LCS | Limbs Coordinate System |
| i-LIDS | Imagery Library for Intelligent Detection Systems |
| ISM | Implicit Shape Model |
| MAP | Maximum a Posteriori |
| MC | Movement Cluster |
| MCM | Movement Cluster Model |
| MOCAP | MOtion CAPture |
| MSEPF | mean shift Embedded Particle Filter |
| NBFV | Next Body Feature Vector |
| NBP | Nonparametric Belief Propagation |
| PAMPAS | Particle Message Passing |
| PC | Pose Cluster |
| PCA | Principal Component Analysis |

| | |
|---|---|
| p.d.f. | Probability Distribution Function |
| PF | Particle Filter |
| PLD | Point Light Display |
| PPF | Partitioned Particle Filters |
| PMRF | Pacific Missile Range Facility |
| PTM | Pose Transition Model |
| PV | Pose Vector |
| ROC | Receiver Operating Characteristics |
| SCFG | Stochastic Context Free Grammars |
| SHMM | Stylistic HMM |
| SIFT | Scale Invariant Feature Transform |
| SIR | Sequential Importance Resampling |
| UKF | Unscented Kalman Filter |
| UPF | Unscented Particle Filter |
| WCS | Word Coordinate System |

## List of Symbols

The below summary of the main symbols is sorted similar to their appearance order. These symbols are grouped per chapter that uses them for the first time, however further chapters may reuse a symbol. Symbols used in local context only might not be listed.

### Chapter 2

| | |
|---|---|
| $\lambda$(A\|x) | the likelihood of observation A given state x |
| $\mathcal{X}$ | a 3D or 2D point |

## Chapter 3

| | |
|---|---|
| $(x_0, y_0)$ | the camera principal point |
| X | a 3D point |
| x | a 2D point |
| $f$ | the camera focal length |
| $\tilde{x}$ | a normalised projected coordinate |
| $\hat{x}$ | an ICS point |
| Q | the camera projection matrix |
| p | the pose vector, a vector of $p^x$ parameters |
| $\mathrm{p}^\phi$ | a partition $\{p^i\}_{i \in x}$ of the pose p |
| $p^x$ | a parameter of the pose p |
| $\phi$ | a partition |
| $\Phi$ | a set of partitions |
| $\mathrm{A}^t$ | the transpose of matrix A |
| $\mathrm{T}_a^b$ | transformation matrix from coordinate system $a$ to $b$ |
| $\pi_r$ | the range prior |
| $\pi_v$ | the visibility prior |
| ${}^s\mathrm{X}_i^j, {}^e\mathrm{X}_i^j$ | silhouette/edge sampling points on generator j and level i |
| O | the observation |
| I | the input colour image |
| E | the edge image |
| S | the silhouette or foreground image |
| $\lambda(\mathrm{A}\|\mathrm{p})$ | the likelihood of observation A given the particle p |
| $\mathrm{chamf}_A$ | the Chamfer distance transform of the binary image A |
| $\mathcal{E}_{i \in T}\, A_i$ | the expectation (*i.e.* mean) of $A_i$, $i \in T$ |
| $\lambda_G$ | a likelihood based on the global pose |
| $\lambda_s$, $\lambda_e$, $\lambda_c$ | silhouette, edge and colour likelihood components |
| $\lambda_l^\Phi$ | a local likelihood based on the parameter partition $\Phi$ |

## Chapter 4

| | |
|---|---|
| $pc$ | a pose cluster instance |
| $mc$ | a movement cluster instance |
| bfv | a body feature vector instance |
| $MC$ | a movement cluster |
| m | a movement vector |
| $l_m$ | the length of movement, *i.e.* the number of the poses producing a movement |
| $n_{\mathcal{C}}$ | the number of clusters |
| $\mathcal{T}$ | the state transition probability matrix |
| $\text{pca}_\text{x}$ | the PCA compressed representation of x |
| BT | the base translation matrix for PCA |
| $\mathcal{C}_x$ | the cluster of $x$ |
| $\mathcal{M}$ | a motion model |
| $\mathcal{M}.x$ | the component $x$ of the model $\mathcal{M}$ |
| $\mathcal{N}(x; m, \mathbf{P})$ | the normal distribution of $x$ with mean $x$ and covariance $\mathbf{P}$ |
| $\text{I}_t^c$ | the input (colour) image of camera $c$ at time $t$ |
| $\text{E}_t^c$ | the edge image of camera $c$ at time $t$ |
| $\text{S}_t^c$ | the silhouette image of camera $c$ at time $t$ |
| $\texttt{Sim}_{\mathcal{C}_c}(\text{m})$ | the similarity of the movement m with cluster $\mathcal{C}_c$ |
| $\mathcal{N}(x; m, \mathbf{P})$ | a normal distribution of $x$ with mean $x$ and covariance $\mathbf{P}$ |
| *mode* | motion mode, with one of the values **pose**, **randompose**, **speed** or **normal** |
| $\sigma_\mathbf{p}, \sigma_\mathbf{r}, \sigma_\mathbf{s}, \sigma_\mathbf{n}$ | stochastic constants of the pose, random pose, pose speed and normal motion modes |
| $\mathcal{L}_l(\text{m})$ | the probability of label $l$ given the movement m |

## Chapter 5

| | |
|---|---|
| $\mathcal{N}(x; m, \mathbf{P})$ | the normal distribution of $x$ with mean $x$ and covariance $\mathbf{P}$ |
| $\mathcal{U}(x; a, b)$ | the uniform distribution of $x$ in range $[a, b]$ |
| $\mathcal{U}\{x; (p_k, \xi_k)\}$ | the discrete uniform distribution of $x$ with probabilities $p_k = \mathcal{P}(x = \xi_k)$ |
| $x \sim q$ | a sample value $x$ drawn from distribution $q$ |
| p | a particle (pose or movement) |
| $\mathrm{p}_t(i)$ | the particle $i$ at time $t$ |
| ${}^{l}_{\tau}\mathrm{p}^{\phi}_{t}(i)$ | the BFV with parameter partition $\phi$ of the pose at $t - \tau$ specified by the $i$-th particle at time $t$ on the $l$-th HPPF level. |
| $\Psi_t$ | a particle set at time $t$ |
| $\phi$ | a partition |
| $\Phi$ | a set of partitions |
| $n_p$ | the number of the particles |
| $w_t(i), \tilde{w}_t(i)$ | weights at time $t$ of particle $i$ |
| $\bar{w}$ | the mean weight |
| $\beta_k$ | an annealing parameter |
| v, w | Gaussian noise samples |
| $n_L$ | the number of level of HPPF current level |
| $l$ | the current level |
| $Partitions_l$ | the set of partitions on level l |
| $p_{\mathbf{p}}, p_{\mathbf{r}}, p_{\mathbf{s}}, p_{\mathbf{n}}$ | the probability of pose based, random pose base, pose speed based and normal motion modes |

## Chapter 6

| | |
|---|---|
| p | a particle (movement) |
| $\mathrm{p}_t(i)$ | particle $i$ at time $t$ |
| ${}_\tau\mathrm{p}_t^\phi(i)$ | the BFV with parameter partition $\phi$ of the pose at $t-\tau$ specified by the $i$-th particle at time $t$. |
| $\Psi_t$ | particle set at time $t$ |
| $\Psi_t^\phi$ | particle set at time $t$ with the partition $\phi$ of the PV |
| $\phi$ | partition |
| $l_m$ | the length of movement |
| $n_\mathcal{C}$ | the number of movement clusters |
| $\mathcal{E}_{i\in T}\, A_i$ | the expectation (*i.e.* mean) of $A_i$, $i \in T$ |
| $\mathrm{O}_t$ | observation at time $t$ |

# Chapter 1

# Introduction

## 1.1 Motivation

### 1.1.1 Public interest

There are multiple interests in video analysis for human behaviour. The most important is public and governmental focus on technological surveillance, either to record and counteract criminal acts, for example in the CCTV cameras deployed in town centres and in public transport termini, or simply to analyse and record behaviours, such as traffic monitoring for congestion charging or road planning, or to observe shopping patterns in a supermarket. Although the extensive operation of static CCTV cameras is obvious, there is no exact, recent estimate of how many cameras are currently installed and their operational scenarios, as summarised by the recent Home Office report [1, p.13]. The best estimate of McCahill and Norris [2] extrapolates from the number of CCTV cameras in Putney to a total of at least 500,000 installations in London, and 4,285,000 cameras in the UK, according to the figures of 2003.

While in the UK there are over 40,000 open street CCTV cameras, probably fewer than 1000 monitor public space across the European countries included in the Urbaneye survey [3]. This limited number questions whether the costly investment is currently managed and exploited effectively. The answer is probably: not.

Given such massive deployment of CCTV cameras, it is just not possible to employ a sufficient workforce to constantly monitor events; an automatic system is needed that, at the very least, could alert an operator to an event in a particular CCTV camera that

requires further scrutiny. This could be as simple as specific motion detection near a perimeter fence of a secure establishment, *e.g.* a motion to climb the fence rather than walk alongside on the public carriageway.

Let us examine a specific case, that of shoplifting. Dabney *et al.* [4] identify that the police have no jurisdiction in retail shops and incidents have to be detected by the in-house personnel. However, shops have a low security budget, and therefore low quality equipment and low paid personnel. The majority of CCTV cameras are in-house systems and the observer, having also other duties, scans the monitor(s) on an irregular basis [2], so that only between 0.1–1% of all cases are observed or reported [5]. Yet, shoplifting detection has important financial implications. Buckle and Farrington [5] compute that about 500 items per week, about 1% of the items taken out from the shop, are stolen from each store by 1% to 2% of the shoppers. According to Dabney *et al.* [4] this number is even higher, and 8.5% of shoppers are involved in shoplifting. Extrapolated to all the shops of a single retail chain in Atlanta, this results in about 2,214,000 incidents per year, of which only 118,529 were detected in reality, equivalent to a 1.7% loss of revenue to the industry. Hence, the financial motivation of surveillance is clear.

Further to crime prevention and detection, behavioural analysis and tracking are important for medical applications, *e.g.* detection of motor disorders, for the supervision of disabled or elderly people, and for home convenience, entertainment systems and for general human-computer interaction. These, with a new range sensors beyond the existing physical input devices (*i.e.* key, button, joystick, accelerometers or LED based positioning), will improve comfort and experience.

### 1.1.2 Scientific motivation

A probabilistic framework for video processing and analysis is motivated first by the probabilistic nature of the input, intermediate representations and output.

The video information has multiple sources of uncertainty. First, at acquisition level sensors and transmitting channels have errors, corrupting pixels or regions; physical dislocation modifies the camera calibration. Further environmental change alter the scene. Objects of interests (*e.g.* humans) may be partly or temporarily occluded by scene objects or other dynamic objects. Also, humans are different, each with individual appearance and dynamics. Hence, exact models are not too complex.

The interpretation of a scene is not unique, people conceive and describe in different

ways what they see. This depends on a priori knowledge about the scene, but also on many additional information that are momentarily perceived. The result is a description different not only on details, but also on the final understanding of the event. As an example, fast walking might be characterised as jogging; or, in a shop a pocket searching action might be considered part of stealing activity if the coming tissue pick action is occluded. These uncertainties require probabilistic modelling.

There is currently great interest in modelling of this kind, particularly using Bayesian analysis, and this motivates the present research. Further, Bayesian analysis allows straightforward incorporation of prior knowledge about the scene in the observation. That is what we, humans, permanently use in the everyday visual understanding.

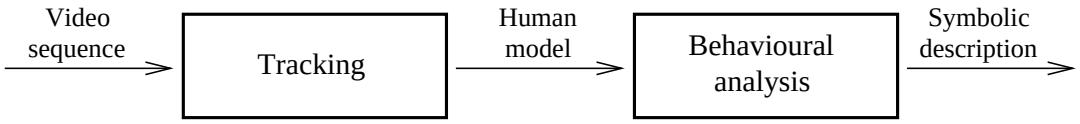



Figure 1.1: Separation of tracking and behavioural analysis.

The thesis separates the visual information abstraction, the computer vision task, from the behavioural understanding, figure 1.1. This is motivated in the next chapter. Any of the two subsystems are replaceable with this modular approach. For tracking a variant of the Particle filter, a Bayesian tracking algorithm, is employed, recently shown attractive in high dimensional parameter tracking. Then, the behavioural system probabilistically analyses the extracted parameters of the model and generates symbols that describe actions and behaviour.

## 1.2 Project overview

The implementation of articulated human tracking and behavioural analysis in a **single** framework is the first challenge of this work. However, the two separable components, tracking and analysis, are clearly identifiable. Since each one is complex, even before they are combined, this thesis can make only a preliminary contribution to some aspects of the problem.

The research is an independent project, funded by the ORSAS and Heriot-Watt University. Since no industrial requirements had to be fulfilled, the research focuses on active and promising topics, however not necessarily with immediate commercial applicability.

Human articulated tracking has not yet been solved effectively using low resolution videos. This was identified as the starting point of the programme.

Furthermore, behavioural analysis from video input has so far been achieved at a coarse level, using blob features, *e.g.* position, speed, shape or internal dynamics. To extend these limited features and to provide detailed and local descriptions, not confined to periodic behaviour, a detailed articulated human model was applied.

## 1.3 The contributions of the thesis

The contributions are in prior information and observation modelling, articulated tracking and behavioural analysis, and the unified tracking-analysis framework.

### 1.3.1 Prior information modelling

Since the subject and environment of the tracking is known, knowledge of the human structure, dynamics, scene and observation provide clues for pose and motion. These are either encoded in the algorithms or learnt by offline training. The articulated model is later used implicitly by the human tracker. The static anatomical model provides the basis of three dynamic models developed to learn realistic human motion. Further, for image information extraction, the model defines multi-component likelihoods. Lastly, the reconstruction of the 3D space for uncalibrated scenes is accomplished by a simple calibration methodology.

### 1.3.2 Articulated tracking

The focus on articulated human tracking, requires a tracker algorithm at the state of the art. This should exploit the hierarchical dependence of some, and the independence of other, parameters given by the anatomical structure. Tracking introduces a multi-modal learnt dynamic model.

### 1.3.3 Action recognition and behavioural analysis

The analysis introduces movement based action recognition. This builds on the dynamic model and, through supervised learning, assigns action labels to movements. Analysis is tested both on good articulated model data and on the tracking algorithm output, evaluating the system as a whole.

### 1.3.4 Unified tracking-analysis framework

Compared to other work, this thesis focuses equally on articulated human tracking and behavioural analysis.

## 1.4 Thesis outline

First, chapter 2 reviews the state of the art in both tracking and behavioural analysis of video sequences. It starts from the psychological literature on the perception of human movement, and then tracking methodologies, prior knowledge and image measurements are reviewed separately, followed by complete systems. The chapter summarises the datasets available for training and testing, and the evaluation methodologies. Finally, it summarises the state of the art and defines the directions of the rest of the thesis.

Next, chapter 3 introduces the static models, including the human articulated model, models for extracting visual information from the images, and the camera model.

The next chapters present the original work on tracking and behavioural analysis. In the temporal order of processing, the tracking precedes the examination of recovered parameters. However, in our framework both behavioural analysis and motion prediction, latter required for tracking, share the same modelling concepts. Therefore these are scrutinised first. In chapter 4, three motion models are introduced, with their training and motion generation algorithms, then the behavioural analysis is defined, which extends one of these models.

The following chapter 5 introduces a variant of a particle filter, able to represent both the hierarchical dependence and the anatomical independence of some model parameters. With the developed switchable motion model and the fine scaled likelihoods, it tracks the articulated motion.

Both sub-systems are tested extensively and independently, and in chapter 6 they are combined and evaluated within the complete system.

The last chapter, chapter 7, draws the conclusions and outlines future extensions for the outstanding problems.

# Chapter 2

# A survey of behaviour and motion recovery

There has been a dramatic increase of interest in the observation and tracking of human subjects through video sequences in recent years. This is directly measurable in a flourishing quantity of research projects and publications. Rather than attempt a complete overview of the state of the art this chapter captures several selected aspects that are relevant to the work to be described. It starts from the psychological background of the behavioural analysis that is preliminary to human pose recovery and tracking. Deterministic and stochastic tracking methodologies are reviewed next, considering their two major driving factors: prior domain knowledge and image analysis techniques. Then, complete tracking systems are reviewed which, compared to isolated algorithms concerned with specific aspects, produce sensible output from the input videos. This is followed by a summary of the available human training and evaluation datasets and by evaluation metrics, focusing especially on those used in this thesis. The chapter ends by summarising the existing problems, and defining those that are discussed in later chapters.

## 2.1 Understanding human behaviour

How humans understand other human behaviour is not yet fully understood, however several sub-systems and mechanisms have been discovered. This insight provides models that are capable of learning and understanding video sequences, resembling human intelligence.

First, behaviour related problems are presented, starting from a psychological back-

ground, followed by the description of behavioural frameworks. Then, the common sense of behavioural terms are examined and defined for later use. Finally, methods of behavioural reasoning are reviewed, with or without intermediate (human) model recovery.

### 2.1.1 Psychological evidence

#### Motion and visual experience

Johansson emphasised with moving *Point Light Display* (PLD) experiments [6, 7] the importance of motion in human perception. The motion of lights attached to human joints, from only two frames and without an observable human shape, is enough to recognise a human figure and to identify the performed activity. However, stationary lights fixed on sitting people do not carry enough information for recognition either of the human or the sitting pose. Johansson argues that projective relations [7] are the foundation for human motion understanding. Humans perceive motion not in Euclidean, but in projective geometry, invariant under perspective transformations, *i.e.* the perceived object is invariant with the viewing angle. The experiments with monocular views generate three-dimensional perception, suggesting that monocular observation for humans is enough to perceive in a 3D space.

The form and motion pathway model of Giese and Casile [8,9] is a neural network based recognition system inspired by biological similarity. A form pathway extracts individual snapshots from movement sequences, while a motion pathway detects action-specific motion patterns. The lowest levels of the two pathways extract features: Gabor filters are used for contour and optical flow for motion detection. The higher two levels assemble the low level information; while on the highest level asymmetric neural connections perform temporal ordering for the final action recognition.

Mather *et al.* [10] use a PLD depicting a moving human figure with a few isolated point sources attached to the major body joints. They conclude that low-level motion detection processes have a major role in human motion detection. The recognition uses the more complete observations, with accumulated motion information. For example, distant joints with complex trajectories (*e.g.* wrist, ankle) provide better recognition than intermediate joints (*e.g.* knee and elbow).

There is no agreement that motion can be used as the only factor for recognition or characterisation of behaviour. Indeed, in either the static or moving camera scenario,

inaction is itself a form of behaviour. Giese and Casile conclude that both form and motion pathways are able to recognise actions, but motion information has some advantages: if trained, then one can recognise degraded motion; action patterns; and recognise the PLD inputs with higher reliability. Prototype based models [9] demonstrate that a well-established neural mechanism achieves robust recognition without static or kinematic human models.

However, the authors used manually preprocessed images for training, actions performed by only single humans, simple scenes, and fixed views. For general application, it would have to be shown that the work can scale to more complex problems.

Further, Jacobs *et al.* [11] performed PLD experiments to show that both visual experience and the execution feasibility of the movement play important roles in human action understanding. The relevance of the motoric factor is suggested by the greater visual sensitivity to possible than to impossible gaits. The visual sensitivity for a movement is dependent on relevance of the motion for the behaviour of the observer. The impact of the motor processes on visual analysis extends across perceptual judgement and situation constraints. Applied to computer vision based behaviour analysis, this suggests the plausibility of a system that examines the scene for motion validity, and if a human dynamic is found then this attracts attention for detailed scene analysis with visual experience based rules. An interesting implication of behaviour resulting from the visual experience [11] is the detection of past positive or negative personal experiences by means of observed behaviour in a given situation, provided by a behavioural database of situation patterns. In summary, motion is the main factor in the perception and behavioural analysis, but shape or form is also important.

#### Prediction and simulation

Human behavioural modelling and understanding is relevant to computer vision since it provides valuable, nature inspired models. Hoogendoorn *et al.* [12,13] propose a model of pedestrian behaviour that is able to simulate observable behaviour patterns. The three-level model successfully resembles human activities performed to approach a specific target. At the strategic level, the departure time and activity sets are decided. At the tactical level the scheduled activities and the area in which they are to be performed are chosen. At the operational level, walking behaviour is refined and the walking route is optimised as function of the current conditions. The behaviour optimisation is formulated as a

cost minimisation that is person dependent, with components from each of the three levels. Operational level cost components are the expected travel time, discomfort due to walking too close to obstacles and walls, walking too fast or too slow, the expected number of pedestrian interactions (discomfort due to crowding), level-of-service, and stimulation of the environment. On the tactical level, deviating from optimal velocity, walking close to another pedestrian and high acceleration or deceleration rate, generate cost. Cost minimisation is solved by dynamic programming techniques, resulting in simulation of unidirectional flows and lane forming in bidirectional opposing flows, with the emerge of realistic structures such as bubbles, strips in crossing flows, and dynamic lanes.

Similarly, within another simulation model, Kukla *et al.* [14] maximise pedestrian flow of real humans in congested areas and explores interactions between pedestrians and other entities. Flow results from the tracks of humans, modelled individually at microscopic level. Each has a goal, one or multiple locations that they want to visit, and each observes and avoids obstructions on their trajectory. The system is based on 30–50 rules of the possible interactions governing the simulation.

Hoogendoorn's and Kukla's works are important since the numerically quantised parameters, costs and rules, define the behaviour. If they are incorporated into automatic behavioural reasoning systems, then once the parameters are recovered, they yield the behaviour, but also predict individual trajectories or the global state of the system.

Other [15, 16] behaviour models focus on human attention fixation. Najemnik [15] simulates human attention with a Bayesian observer model. The resulting model resembles human characteristics: it fixates at the maximum posterior probability, has moderate saccade length, tends not to fixate where it recently fixated, and makes long saccades into regions with higher probabilities.

Summarising these results, both motion and experience are involved in perception, however motion provides more detailed information; for humans, joint location is performed quickly and effectively using only a few points. Models of behaviour provide mathematical parametrisation that together with recognition, allow prediction.

### 2.1.2 Behavioural systems: services and requirements

Looking at the needs of a behavioural system, Krumm *et al.* identify [17] the basic services (table 2.1) that an intelligent environment should offer. Provided this, the system has to satisfy the several requirements listed in table 2.2. Event triggering is the most generic

task, being the main interest of surveillance. An obvious example is to raise the alarm in response to a forbidden position or behaviour. Locating devices, invoking user preference and user assistance have their application in convenience type applications. The complex requirements of a behavioural system demand good organisation, with multiple tasks, and optimised speed and robustness.

| Event triggering by positioning |
|---|
| Locating the right device to use (for something) |
| Invoke a particular user's preferences |
| Understanding behaviour in order to assist |

Table 2.1: Behavioural systems intelligent services [17].

| Maintain location and identity information |
|---|
| Run at reasonable speed |
| Work with multiple people |
| Create and delete people as they appear and disappear |
| Cover whole area (with multiple cameras) |
| Obtain perspective camera view (hard with aerial views) |
| Extend working time |
| Tolerate occlusions and different poses |

Table 2.2: Behavioural systems requirements [17].

Crowley [18] presents a general modular, process-based software architecture for real time observation of human activity. An element of the system is the perceptual process, composed from a set of modules controlled by a supervisory process. To perform a task (*e.g.* face and hand detection), assemblies of processes result in composite entities and cooperating process federations that are controlled by a meta-supervisor, launching and controlling processes or launching other meta supervisors.

The conclusion to be drawn from this is that a behavioural system requires such an organised, modular design. Since the human brain performs quickly and efficiently all the above intelligent services and the relevant subtasks, it is important to consider its architecture. Whether it is advantageous to base machine on human perception is an open question [19], but one should at least be aware of the possibility.

### 2.1.3 Psychologically inspired system models

Lee and Mumford [20], driven by the current interest in Bayesian and Particle Filter frameworks, model visual perception hierarchically and show that early visual neurons from the lowest visual level, the V1 area, process from local to global, while the higher-level, the IT neurons, act in a coarse to fine manner, first extracting generic information

(*e.g.* the gender before the face). These two areas interact continuously and constrain each other.

Hierarchy is suggested also by the increase in the receptive field size of the V1, V2 and IT visual areas and the number of active neurons representing a hypothesis of possible recognition outcomes. While 10% of the total V1 neurons represent an individual hypothesis, this percentage increases to 20% and 100% for V2 and IT neurons respectively, reducing the uncertainty in the higher neural levels with successive convergence of visual information. By means of neural resonance, multiple hypotheses at higher levels collapse, and the feedback to the lower levels limits the explosion in the number of hypothesis. At the highest level only a single hypothesis is kept, while lower levels maintain alternatives that emerge if new observations sustain them.

Lee and Mumford [20] also argue for generative models, with top-down reasoning constrained by errors provided by the bottom-up pathways showing low activity when the generative model matches the data. They show that alternative competition models are not powerful. Generative models are supported by the illusory contours generated in the V1 neurons. Here, the simplest explanation is chosen from multiple possible hypotheses, even at the extra cost of hallucinating a contour. This agrees the Gestalt organisation law of Prägnanz that selects the best, simplest and most stable shape of a geometrically organisation from a set of possible explanations [21, p.127].

Mather *et al.* [10] prove the existence of low level, biological motion detectors, performing over short inter-frame differences, while Hosoya *et al.* [22] show that simple image processing is performed already in the salamander and rabbit retina. Physical measurements of the transmitted stimuli attest adaptation of the retina with the background pre-selecting the useful information, but also performing signal compression in order to reduce the information transmitted to the brain, this being comparable with predictive coding. Similarly, Jacobs and Werblin's [23] salamander retina experiments show that edge detection is performed at the low level of the retina cells.

Bregler [24] suggests that low level distributions propagate towards the higher levels, and that it is only at the top layer that hard decisions of recognition are taken. The Bregler guidelines are: no early commitment to a specific hypothesis; beside bottom up flow, the higher level hypothesis should be able to disambiguate lower level estimates; low computation and representation costs; mid and higher level models should be human readable.

Physiological evidence and biological experiments conclude that visual information processing starts at the low level of the retina, and in a hierarchical manner builds up towards abstract representation, while each level reduces the complexity. The information flow is not only bottom-up, but is also top-down, using generative models. The hierarchical organisation of such a system requires multi-level modelling. This suggests the need for probabilistic modelling where multiple hypotheses can exist on multiple levels and the decision is delayed to the highest level. Additionally to bottom-up, top-down interaction between levels may feed back information and enhance previous inferences. This type of model, with the possibility of distributed low level processing allied to high level generative processes is not incompatible with a distributed sensing and processing network, which is the direction of current technological development.

This class of behavioural analysis does exist in hierarchical and stochastic model design. For example, Nagel suggests [25] a layered organisation. At the lowest level the *change* reflects a sensory signal, observable from the image. A *change* is assembled by a priori knowledge into an *event*. A *verb* describes some *event* or absence of activities, while *history* expresses an extended sequence of related activities.

In contrast to Nagel, the model of Green and Guan [26,27] has four abstraction layers: *dyname*, *skill*, *context* and *activity*. The basic element is the *dyname*, equivalent of the *phoneme*, frequently used is voice recognition. For the recognition of the dynames, *Hidden Markov Models* (HMMs), a basic tool from voice recognition, are used. The higher levels are defined in a Bayesian manner and modelled similarly with HMMs.

The motion analysis and recognition framework of Sanfeliu and Villanueva [28] has six levels. The model builds two and three dimensional information from image features, used for conceptual and behavioural level analysis. Most levels communicate in a hierarchical manner, but more complex interactions between levels are also present. The proposed model is reasonable, but without a complete implementation cannot yet be validated.

### 2.1.4 Pose, movement, action, activity, behaviour and gesture

The terms pose, movement, action, activity, behaviour and gesture are often used interchangeably in the literature and in everyday language, defined only by the language context, the speaker, or the required emphasis. This section quotes the common dictionary definitions and states explicitly those used in this work as numbered definitions.

The posture or pose is *the position or bearing of the body whether characteristic or*

*assumed for a special purpose; the state or the condition at a given time especially with respect to capability in particular circumstances* [29]; *the way in which someone usually holds their shoulders, neck and back, or a particular position in which someone stands* [30].

In this work,

**Definition 1.** A *pose* is a static local or global body configuration.

The pose defines the configuration of the whole body, of one or more body parts without intentional meaning.

A *movement is the act or process of moving; change of place or position or posture; a particular instance or manner of moving action* [29]; *a change of position* [30].

**Definition 2.** A *movement* is a short, continuous sequence of poses, having no intentional content.

An action is *an act of will; the accomplishment of a thing usually over a period of time, in stages, or with the possibility of repetition* [29]; *a physical movement; the way something moves or works; things which are happening, especially exciting or important things; the process of doing something, especially when dealing with a problem or difficulty* [30].

**Definition 3.** An *action* is a short sequence of poses (*e.g. leg rising, arm still*). It is usually, but not exclusively, defined by one or some body parts.

An action and a movement are similar, except that actions have an intentional content.

An activity is *the quality or state of being active; vigorous or energetic action: liveliness; natural or normal function; a process that an organism carries on or participates in by virtue of being alive; a process actually or potentially involving mental function; a pursuit in which a person is active* [29]; *the work of a group or organisation to achieve an aim; when a lot of things are happening or people are moving around* [30]. Therefore the definition of activity used here is:

**Definition 4.** An *activity* is a symbolic characterisation of the body over a limited time, bearing an intention.

Gesture is *a movement, usually of the body or limbs, that expresses or emphasises an idea, sentiment, or attitude; the use of motions of the limbs or body as a means of expression* [29]; *a movement of the hands, arms or head, etc. to express an idea or feeling; an action that you take which expresses your feelings or intentions, although it might have little practical effect* [30].

**Definition 5.** A *gesture* is the expression of an emotional content by a short pose or movement of the limbs or the whole body.

A gesture is related to an action and activity since all have intentional content, however it has additional emotional component. This emotional content is as yet ignored by this study, and therefore actions and activities will suffice in this work.

Behaviour is *acting in a particular way, or to be good by acting in a way which has society's approval* [30]; *the manner of conducting oneself; anything that an organism does involving action and response to stimulation; the response of an individual, group, or species to its environment; the way in which something functions or operates* [29].

**Definition 6.** Then *behaviour* implies public approval or disapproval of an activity.

| Term | Dynamic | Extended time | Intention | Full body | Direct attention |
|---|---|---|---|---|---|
| Pose | No | No | No | Yes | No |
| Movement | Yes | No | No | Both | No |
| Gesture | Both | No | Yes | No | No |
| Action | Yes | No | Yes | No | No |
| Activity | Yes | Yes | Yes | Yes | No |
| Behaviour | Yes | Yes | Yes | Yes | Yes |

Table 2.3: Pose, movement, gesture, action and activity. Summary of dynamic, time, intention, description level and attention orienting properties of pose, movement, gesture, action and activity.

The similarities of pose, movement, action, activity and behaviour are summarised in table 2.3 from the perspective of the time (being static or dynamic); length of time (short or extended); if it has any intentional meaning; if it is described by full body or just body parts; and if it requires further attention (*i.e.* of an operator). Table 2.3 suggests that pose and movement are the two basic blocks, neither of them has intentional content, added later by gesture and action. Only the pose is static, while activity and behaviour have an extended time span.

**Definition 7.** *Behaviour analysis* is the process to detect if an activity has been performed and if this activity is approved or not.

The above definitions imply that behaviour recognition starts from the detection of the lower level concepts, assembling them into activities, that are then assessed context dependently as positive or negative.

The distinction between activity and action is frequently blurred in everyday communication. Action, as defined above, can occur as part of an activity, or as a standalone activity, if this is the highest level of understanding required. For example, walking is an activity, if the set of activities are running, jumping and walking; but, if the activity in focus is walking with one's hands up, then walking is an action required for a more complex activity. Similarly, reaching (something) is an activity, however if it is part of a catch/throw activity, then it is considered an action. On this basis, action recognition results in activity as either regarding the action over an extended time and attributing it with an intention, or the concurrent presence of multiple actions is considered an activity. Therefore, activities are assembled from one or more actions, and actions provide specific, extended understanding detail on an activity (*e.g. walking with raised left arms*).

Bobick [31] considers movement, action and activity as the three abstraction levels of perception. Movement is a continuous motion easily characterised by the trajectory in some configuration space (*i.e.* dynamics of the motion, defined by speed, acceleration). The movement has unique definition, while activity is a statistical temporal combination of movements. In addition, the activity understood in its context defines the action. For Bobick, the activity includes the contextual dependence, while definitions 3 and 4 allow actions and activity relative to an environmental context such as picking up a ball (action) or meeting someone (activity). However this requires additional context knowledge, the object of the action or activity.

İkizler and Forsyth [32] use acts and activities as basic blocks of the recognition. Acts, similar to the actions above, are HMMs with a low number of states. Similar states are interconnected in larger HMM representing activities. The acts and activities are modelled at a multiple limb level. Green and Guan [26] have four abstraction levels: the dyname, skill, activity and context. The first three correspond to movement, action and activity while the context is a prior knowledge about the conditional probabilities of the skills.

Bregler [24] employs a bottom-up method, detecting low level features and trying to assemble them into high level descriptions. On the second level, a HMM with a forward-backward algorithm is used for the best segmentation into gestures of the simple features category called *movemes* (similar to *phonemes*). Recognition is performed only at the top level analysis.

In the terminology of this thesis, pose, movement, action and activity are considered as the abstracted layers of understanding. The process of inferring any of the four is referred

to as behaviour analysis. Poses and movements are physical behaviours, while actions and activities directly carry social value that needs to be detected, observed, valued or identified.

### 2.1.5 Behaviour understanding and human model recovery

In contrast with İkizler and Forsyth [32], who categorise human recognition using three methods, temporal logic, spatio-temporal templates and dynamic models, the primary discrimination factor this review considers is the intermediate abstraction level. The first class of methods directly analyse visual data while the second class first recover detailed (*i.e.* articulated model) or raw (*i.e.* position, orientation or speed) human body parameters and then analyse these parameters to recover the behaviour.

#### Behaviour understanding without human model recovery

Rittscher *et al.* [33, 34] use features of an image sequence's *space-time (XYT) cube*. They suggest that a learnt distribution of skew vectors, modelled by a multivariate Gaussian distribution, can be used to identify an action. These features have no explicit relationship to human body parts, although clearly the space-time envelope is dependent upon human anatomy. They use the epipolar slices of Cipolla [35], but not only the slice at the camera height. However, points from other parallel planes change their plane as the human depth changes. While in [35] sampling at the constant camera height ensured that points of a person walking back and forth remain on the same epipolar line, this is not true in the general scenario assumed in [33, 34].

Howell and Buxton [36,37] use a *Time-Delay Radial Basis unction Network* with a 126-coefficient feature vector input, formed as a result of differencing and thresholding Gabor filters. The network is fast, however it is only applied to limited databases. Since the method is view-dependent and requires segmentation of the scene into individual targets, training for general recognition is too complex in our view. The method of Masoud and Papanikolopous [38] is inspired by Johansson's experiments. They suggest that tracking is possible using direct image features. Therefore, an infinite impulse response filter is applied to a stack of silhouettes, representing the history of image motion, and the result is compressed by *Principal Component Analysis* (PCA) into a manifold. Metrics on distances between manifolds express the similarity between actions. The method performs well on

large action databases, however it uses assumptions such as a fronto-parallel view and known human position and dimensions. This would make the method inapplicable to most surveillance applications with complex scenes, human-machine interfacing or intelligent environments.

Wang *et al.* [39] use local, low-level visual features based on moving pixels, resulting from the thresholded intensity differencing of successive frames. In $10 \times 10$ blocks, pixel position and direction features are quantised with a *codebook* resulting in *action words*. These action words are analysed with Bayesian hierarchical techniques for video summary generation, video segmentation and abnormality detection of distant traffic scenes. Local, spatio-temporal features are also fed into a trained support vector machine that recognises actions accurately against a low complexity background [40]. Since the background is not removed, weaknesses of this approach are that it cannot deal with complex scenes and unseen viewpoints.

Lv and Nevatia [41] use a sequence of key silhouettes, representing *key poses* to describe actions. The actions are recognised by Viterbi path search. Weinland *et al.* [42] use space carving with silhouettes to reconstruct the 3D volumetric representation of the body, of which the motion history provides directly the features used in recognition. This method is on the border between tracking and tracking-free recognition, since it does not need articulated model parameters, but reconstructs in each frame the 3D volumetric representation that is tracked by repeated detection.

The recent use of *space-time cube* analysis to detect similarities of actions, extends two dimensional (*i.e.* space) into three dimensional (*i.e.* space and time) correlation [43–45]. Shechtman *et al.* [43] define the similarities of the time-space patches characterised by rank increase of the time-space gradient, computed by PCA of the Gram matrix of the patch gradient. Highly correlated patches describe similar actions. Boiman [45] uses sets of similar patches from different sequences to explain regions of the video. Regions that cannot be explained by other patches of the same sequence are irregularities of the video. Trained only with normal behaviour (*i.e.* walking and jogging), the algorithm recognises every other distinctive behaviour (*e.g.* jumping, carrying, laying, *etc.*; see figure 2.1). Similarly, Blank *et al.* [44, 46] extract space-time saliency and orientation features (*plateness, stickness and ballness*) of the silhouette history images, integrated over the whole space-time volume. The method gave good recognition rates with several, mostly periodic, actions, but not for the bend action.

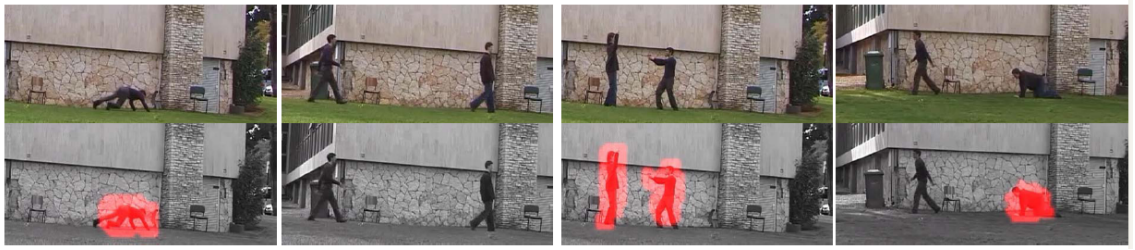


Figure 2.1: Detecting irregularities (not trained behaviour) with space & time correlation [45].

The main advantage of these last two approaches compared to previous space-time volume methods is that they use local patch analysis; these patches are later assembled into regions. Local analysis, in contrast to the larger XYT cubes that include background, or rely on good background subtraction, aim to be more robust, with less view dependence and greater flexibility. Space-time analysis is very attractive since it avoids the computational expense of tracking, however its global success is doubtful. In the absence of tracking, the raw image is processed without layered information abstraction, which is a feature of human vision, as shown in section 2.1.3. There are further arguments against these methods. Simple statistical features defined directly on the image or on the image-time space, cannot capture the complexities of 3D human motion, and are view dependent. Given the variety of human activities, a large set of statistical measurements would be necessary. Finally, background clutter significantly changes the *blind* statistical measurements. However, as camera and scene are generally fixed, background subtraction can mitigate partly this disturbance. To further minimise these factors would require analysis only at the location of the event, but this localisation (*e.g.* by tracking) is exactly what the proponents of this approach desire to avoid. To date, these methods have used statistical measurements compiled from the video data, and analyse a limited number of activities. There is no evidence yet of scalability with increasing number and complexity of activities, environment, and viewpoint.

In summary, although a full human model is not recovered, it is clear that basic tracking is still required for localisation and extraction of silhouette or regions of interest. Tracking-less analysis is not really feasible; at the very least such a system should employ least basic blob tracking to define the space-time envelopes of interest.

**Behaviour understanding with human model recovery**

The explicit recovery of a humanoid model, and the independent analysis of the recovered model parameters, separates image analysis, *i.e.* handling images and videos and extracting compact information, from data analysis and artificial intelligence, *i.e.* abstracting the extracted information into a symbolic description.

Mühlenbrock *et al.* [47] use *Bayesian Networks* for recognition with automatic MAP learning. Like many other systems of this kind, they learn only a few actions, and once these are learnt, it is not possible to update or manually override the learnt parameters. Remagnino *et al.* [48] also use Bayesian networks to infer the description of the scene. The reasoning has two levels: the *behaviour* agent operates at the low, image level, and describes the object's behaviour by its dynamics and trajectory, obtained from location, heading, speed, and trajectory nodes. The *situation* agent at the second level decides the situation of one or two objects, using the parameters extracted by the behaviour agent. In this framework, the authors successfully augment surveillance scenes of moving or parked cars and walking persons with symbolic descriptions such as *Pedestrian x is moving towards car y*, *Car z is parked in the parking-lot*, *etc.* The resulting extracted situations or behaviours are impressive, however they are again limited to simple scenarios.

*HMMs* are frequently used to detect simple actions or as low level detectors. For example, Brdiczka *et al.* [49] recover a conversational group from multi-channel speech detection. Nickel and Stiefelhagen [50] model the three phases of a pointing arm gesture with a HMM in order to detect where a user points while interacting with a house-hold robot. Ivanov and Bobick [51] also use a backward looking HMM to detect short-length events.

The W4 system [52] is based on empirically adjusted *rules* and employs computationally simple methods to detect simple activities and strange behaviour (*i.e.* carrying an object). Fuentes *et al.* [53, 54] apply a rule-based method for special event detection (*i.e.* unattended luggage, a fall, hiding, vandalism and fighting) based on reasoning about 2D blob features. The features used are centre position, blob splitting, merging, appearing and disappearing, changing speed and size.

A common thread in much of the research on intelligent environments is that the methodologies used stem from the *early artificial intelligence vision* systems and attempt to extract a symbolic description from low level movement detectors. In other work,

Brdiczka [55, 56] provided services while minimising disruption by perceiving user activities and identifying needs, adapting and developing automatically for a smart-office environment, modelled by an *ID3 decision tree*. By dynamically changing the decision tree (*i.e.* splitting and deleting nodes), the system is able to learn and adapt the reaction. Supervised learning was used for a limited, manually encoded test scenario and so the scalability of the method is not proven. Brand [57] describes activities with manipulators. Objects are detected as blobs, and the position and motion of the leading edge, area of the blob and changes in front or behind the leading edge define the basic events (*i.e.* appearance, disappearance, fusion, deflation, flash, acceleration) that according to a manipulator grammar define high-level actions (*i.e.* add, touch).

Once basic symbols are extracted from a tracking process, *Stochastic Context Free Grammars* (SCFG) [58] can generate higher level descriptions since they provide flexible and simple rules for compounded symbols, describing in a natural way human actions and behaviour. For example, such a set of rules, describing a *fight* may be written as

$$
\begin{aligned}
&\text{FIGHT} \rightarrow \text{FIGHT FIGHT} && [0.20]\\
&\text{FIGHT} \rightarrow \text{KICK}_R && [0.20]\\
&\text{FIGHT} \rightarrow \text{KICK}_L && [0.20]\\
&\text{FIGHT} \rightarrow \text{PUNCH} && [0.40]\\
&\text{KICK}_R \rightarrow \text{KICKSTART}_R\ \text{COLLAPSE}_{human}\ \text{KICKFINISH}_R && [0.50]\\
&\text{KICK}_R \rightarrow \text{KICKSTART}_R\ \text{KICKFINISH}_R && [0.50]\\
&\text{KICKSTART}_R \rightarrow \text{static}_{RLeg}\ {\text{RISE}_{RLeg}}^* && [1]\\
&\text{KICKFINISH}_R \rightarrow {\text{FALL}_{RLeg}}^*\ \text{static}_{RLeg} && [1]\\
&\ldots
\end{aligned}
$$

with the terminal $\text{static}_{\text{RLeg}}$ inferred directly from the model parameters. For the right kick, $\text{KICK}_R$, the two rules are similar, but the second requires in addition to the leg dynamics a predicate that someone collapses (*i.e.* the result of the kick). For real situations, complex production rules, also learnt from training, are required.

For high level analysis PRISM [59] provides a plausible framework. It is a symbolic-statistical modelling language, built on extended Prolog with additional stochastic predicates. The most important predicate is the random binary switch (`bsw/3`), a random variable set to a 0 or 1 value and with distributions learnt by example based training with

PRISM predicates. A set of predicates define the statistical model, based on the distribution of `bsw/3` trained by *Expectation Maximisation* (EM). For reasoning, other predicates evaluate and sample the learnt distributions. PRISM covers a large class of statistical models, including Bayesian Networks, HMM, and *Stochastic Context Free Grammars* (SCFG).

Ivanov and Bobick [51] use SCFG on two levels, first simple actions are recognised (mainly by HMMs), then SCFG assembles the data using stochastic rules, therefore not only are the rules stochastic, but also the input signals. Their tests include drawing a square in the air by hand, with HMMs to detect left-right, right-left, up-down and down-up primitives; and the grammar of musical conduction in 2/4 and 3/4 time. They have also applied their methods to surveillance event monitoring of cars, and of people performing enter, exit, drop-off, and pick-up events. Their multiple applications show the flexibility of SCFGs, despite of the missing objective evaluation, manually designed rules with estimated, not learnt, production probabilities. Ivanov and Bobick suggest that a combined stochastic and symbolic approach is applicable because there is insufficient data with only parts of the full information being present; there is semantic and temporal ambiguity, and different rules have different lengths. Although their architecture has a known structure, it is difficult to construct a learning system even when an explicit, a priori model is known. For PRISM and for the SCFGs the time model is implicit and is given by the order of predicates in the production rule. This is a flaw, since some actions need backward access to events performed in the (relatively) distant past.

Although behaviour analysis can be achieved without tracking and pose recovery [44–46], the drivers for tracking, also formulated by İkizler and Forsyth [32], are to achieve view independence of behavioural analysis, based on training with reliable ground truth data acquired by motion capture systems. If separated from low level image analysis, then occlusion, self occlusion, illumination and environmental change issues are solved independently. For the spatio-temporal methods, tracking or detection is still required to localise the area of attention on the human subject, even if the methods are simple. On the other hand, high level behaviour analysis based on extracted model parameters generally applies a Bayesian framework (Bayesian networks, SCFG) in order to incorporate human-readable behavioural prior knowledge. At the lower level, detection is based on HMM or ad-hoc detectors. Such task separation corresponds to the physiological abstraction into hierarchical levels, discussed in section 2.1.3.

### 2.1.6 Conclusions

Human behavioural understanding with computer vision has been solved only for specific applications. Without an intermediary representation of the human, algorithms are successful only with problems that have been rigidly defined a priori. Intermediate model-based techniques, motivated by human psychology, are more flexible but have high complexity.

## 2.2 Methodologies for pose recovery and tracking

We have discussed arguments for isolated processing of image sequences to extract information as a primary step of behaviour analysis in section 2.1.5. Tracking to recover human pose within each frame provides an intermediate abstraction level. Consecutive frames provide continuous pose changes that can be exploited by the tracking algorithm. In this section, existing tracking methodologies are reviewed critically, by general description, theoretical background and with practical examples.

**Definition 8.** In this thesis, *tracking* is considered to be the process of continuously estimating the current state parameters of the system model, using external measurements, generally in the presence of environmental noise and clutter.

For Ristic *et al.* [60, p.14] tracking is a form of data processing to form and maintain tracks; a track is a sequence of target state estimates up to the current time. Tracking is an estimation problem [61, p.115], therefore predictors, filters and smoothers are relevant, each estimating the state of a system. They differ in the temporal range of the input data used for estimation: predictors use only observations *prior* to the current time of estimation, filters use observations *up to and including* the current time, while smoothers use observations *beyond* the current state to be estimated. This defines tracking as a filtering problem. Although filtering is the main task, tracking is strongly related to the other two estimators, since filtering (Kalman, particle, *etc.*) uses a prediction step to reduce the search space, while smoothers are able to incorporate future observations for the current state when a delay of the estimate is acceptable.

The two major classes of tracking, deterministic and stochastic methodologies, are considered next.

### 2.2.1 Deterministic tracking

Deterministic tracking is characterised by systematic global or local search, conducted exhaustively over a pre-defined range. It has the advantage of robustness, however if the parameter space is large then it entails an explosion of demand on processing resources.

#### Tracking by detection

Repeated detection continuously applies the same detection algorithm to each frame. Previous detection results are either ignored completely [17, 62–64] or used to partly restrict the detection space [52, 65]. The detection results in one or a set of targets, and it needs to solve for the identity of each object in each frame, the data association problem.

Krumm *et al.* [17] apply repeated detection of consistently moving, human-like blobs. The identity of each detected person is maintained by a colour cube histogram of each cell of the $10 \times 10$ map. The mobile robot of Böhme *et al.* [66] tracks humans with a Fuzzy-Minimum-Maximum operator, subsuming colour and head-shoulder contour clues. For initial detection, the face provides an additional clue. The tracking suffers from illumination and background changes, therefore the performance is not satisfactory for real applications. To solve this, an extra sonar and microphones were used for source localisation, adding to the cost with expensive hardware.

Chamorro-Martínez *et al.* [67] segment the hands as a coherently moving objects, resulting from the frequency analysis of a spatio-temporal cube. Similar to other spatio-temporal approaches, unwanted objects, a cluttered background or complex object articulation may ruin the method, therefore its generality is limited to simple scenes. Pixel motion coherency is an important segmentation clue. Rao and Shah [68, 69] detect the position of the hands as a centroid of the fastest moving skin colour area, using a skin colour histogram built offline with manually classified skin areas. Cheng and Chen [65] use wavelet decomposition to smooth the images, and on the third wavelet level, the differencing results in the moving regions. The disconnected regions are post processed with a morphological closing operation, while detected blobs are analysed in a bottom-up manner.

Leibe *et al.* [64,70,71] apply an *Implicit Shape Model* (ISM) to detect objects by means of an appearance codebook. The learnt appearance patches are encoded relative to the centre of the object, voting for the object identity and centre location. The method is fast

and highly effective [71] and concludes that for detection, the best shape context (*i.e.* the edge orientation histogram) is obtained from real edges, in contrast to silhouette contour edges. These also perform better than a local Chamfer descriptor, outperforming the global shape descriptors. Seeman *et al.* [72] extend the above ISM approach by defining global descriptors and their joint probabilities, resulting in a generative object model that can detect both classes (*e.g.* pedestrians) and instances of classes (*e.g.* individual pedestrians; see figure 2.2).

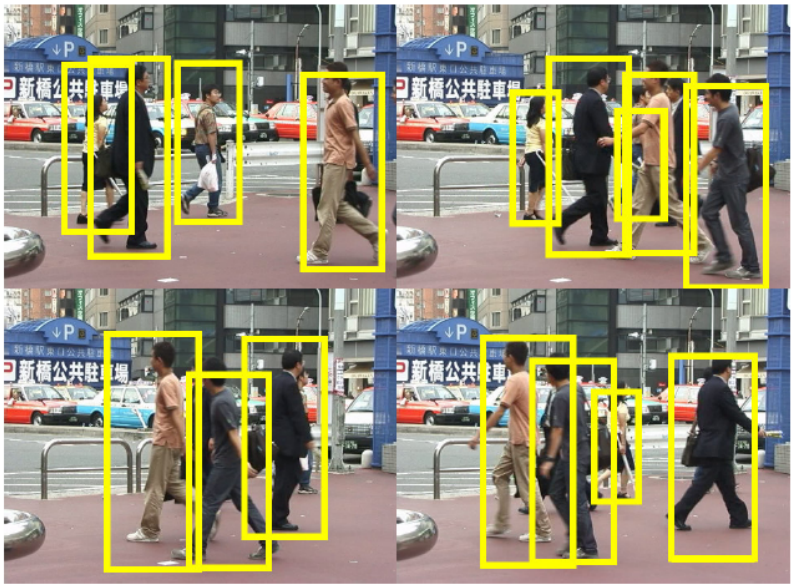


Figure 2.2: Codebook based pedestrian detection with ISM [72].

Boosting is a method that obtains an accurate and robust classifier by chaining together rough classifiers with limited accuracy. Adaboost [73] learns a set of weak classifiers from a pool in sequential order. Each new classifier is chosen so that it best classifies the as yet worst classified samples. Viola and Jones [62] use a set of cascaded feature detectors to reject iteratively non face-like objects. The simple processing steps required for each detector and their cascaded design proves to be very efficient and in each frame the repeated detection results in robust face tracking. Similarly, Viola *et al.* [74] build a pedestrian detector using trained cascaded detectors, with decisions based on intensity and motion information (*i.e.* subtracted consecutive frames). With 4 *frames per second* (fps) on a 2.8 GHz P4 processor, the detection is fast, with a low false alarm rate, and successful with small images of persons. Adaboost was successfully integrated into IBM's S3 surveillance system [75] for face and hence human detection, and is now one of the

most widely cited and applied approaches.

Gavrila use [63] coarse-to-fine and hierarchical search in a hierarchical template tree with node decision on edge orientation of traffic signs. Gavrila and Philomin [76] extend this approach, for learnt pedestrian templates. K-mean clustering at each level of the hierarchy and simulated annealing is used to learn the template hierarchy from a database of about 1000 pedestrian shapes with their pairwise dissimilarity computed using a distance transform. The detection rate is 75–85% at a processing frequency of 1–5 Hz.

The W4 system of Haritaoglu *et al.* [52], mentioned already in the context of rule-based behaviour analysis, uses a heuristic human model from the horizontal and vertical projection of the foreground regions. A second order motion model predicts the probable locations and reduces the search space in each frame. The new initial position is the median coordinate of the blob pixels at the previous location. W4 tracks individual body part positions by repeated and independent heuristics of silhouette feature points.

Delamarre and Faugeras [77] recover articulated human models from three views, driven by the forces between the estimated projected silhouette and the observed silhouette, using results from robotics to compute body part accelerations from multiple 2D forces dragging the object towards the real position.

To summarise, in a wide range of approaches tracking is performed by repeated detection. Such tracking generally requires continuous discovery of the screened model parameters. The detection algorithm is heuristic and dependent on the experience of the designer, on the specific features of the object, and on the environmental conditions. The detection criteria are variable; features such as contour [66], [52] colour [67–69] or movement [17,68] have the advantage of fast implementation. On the other hand, a learnt model [72,74,76] offers robustness and flexibility. Models such as Adaboost suggest a better detection rate, with less in built features, but with the cost of the necessary training.

### Feature matching

One of the earliest feature point trackers was based on the similarity of affine transformed intensity regions [78] for registration and calibration; based on the intensity map and its first order derivative, disparities between the model in the preceding frame and possible targets are computed. Repeated iterations converge towards the correct position of the

model $G$ in the new frame $F$ given by the update:

$$\Delta y = \frac{\sum_x \left(\frac{\partial F}{\partial x}\right)^T [G(x) - F(x)]}{\sum_x \left(\frac{\partial F}{\partial x}\right)^T \left(\frac{\partial F}{\partial x}.\right)} \tag{2.1}$$

Though it can be applied to large regions, equation (2.1) is adopted more frequently for tracking feature points within a small local neighbourhood, such as corner points or edgelets. Shi and Tomasi [79] define good features to track, corresponding to real object points, as those with high dissimilarity using affine motion model for the underlying image changes. These define the *Kanade-Lucas-Tomasi* (KLT) feature tracker. CASSANDRA [80] uses the KLT tracker for human tracking, and the energy of the feature points results in a behavioural analysis.

The *Scale Invariant Feature Transform* (SIFT) of Lowe [81] uses relevant and stable features for object detection and recognition. SIFT is scale and illumination independent, at least for linear illumination change. It has some immunity to rotation and translation (affine transformation) changes. SIFT features are scale-space extrema: at multiple scales difference of Gaussians minima and maxima are detected as proposal feature points; low contrast points, including edge responses, are rejected. Modes of edge orientation histograms are used as descriptors, computed in the areas around the feature points. Histograms of image regions, using image feature point location, scale and orientation are built, to facilitate recognition. The aforementioned illumination changes are reduced by histogram normalisation. These present a good alternative to KLT features.

#### The mean shift tracker

The mean shift algorithm is a nonparametric statistical method to find the nearest mode of a point sample distribution. This algorithm has been adapted as an efficient technique for image segmentation [82, 83] and object tracking [84, 85]. The justification for its use as a density estimation-based non-parametric clustering method is that the feature space can be regarded as the empirical *probability density function* (p.d.f.) of the represented parameter. Dense regions in the feature space correspond to local maxima of the p.d.f., *i.e.* the modes of the unknown density. Once the location of a mode is found, the associated cluster can be delineated based on the local structure of the feature space.

The Mean shift has been modified by several heuristics for occlusion handling [86], better foreground and background separation [87, 88], and high velocity object tracking [89].

It tracks non-rigid objects such as humans and faces as blobs. It is inherently only a blob tracker with the area size and shape defined by a kernel function. For human tracking in a highly cluttered space (a subway platform) the mean shift was used [84] with good tracking performance. Comaniciu *et al.* suggest that geometric constraints and background subtraction substantively enhance the algorithm. Such enhancement, for example including a foreground mask in the kernel $k$ by replacing the Epanechnikov kernel with a Chamfer distance transform, improves the tracking, as shown by Chen *et al.* [90]. Similarly, Porikli and Tuzel [88] use foreground and skin colour for a more directed kernel, that counts only foreground pixels. Unfortunately, these methods do not have analytic proof of convergence, although tests shows faster convergence then the original Mean shift [90].

Collins [87] uses two interleaved mean shift procedures: one tracking in the location space and one in the scale space. The features of the tracked images are difference of Gaussians that from the image extract, on multiple scales, blob like areas. The interleaved multi level approach wisely reduces the problem of searching in three dimensions to a pair of two-dimensional searches, that mean shift was designed to solve. The author presents robust, scaling resistant results for non-occluded tracking of a single human.

Further, with continuously maintained identity map, Lerdsudwichai *et al.* [86] extends the basic mean shift to handle simple or multiple occlusions. When multiple objects occupy the same region of the map, the maximum value of the similarity function can judge which object is visible and which is occluded.

Tracking an object with high velocity can be performed by including a Kalman filter for prediction [89] or combining the mean shift with other probabilistic prediction methods from section 2.2.2.

#### Active contour trackers

*Snakes* were proposed seminally by Kass *et al.* [91] to represent flexible contours, and since have received strong attention from the image processing community. Snakes, modelled with B-splines, were successfully used by Baumberg and Hogg [92] for human contour tracking and were then included in the Integrated Traffic and Pedestrian System [93] and in the Reading people tracker [94]. Unfortunately, the B-spline model fails for the large variance of contour generated by the hands. The level set tracker [95] is an alternative to snakes. It models the object with the zero level set of a function, however the advantage of uncontrolled modelling of independent regions (*i.e.* split and merge of regions) in tracking

humans is a flaw since it can divert the tracking.

### 3D structure from 2D

Leventon and Freeman argue for the use of strong prior knowledge, and show that this is useful for motion generation and 3D pose recovery "*as one watches a film or video of a person moving, one can easily estimate the 3-dimensional motions of the moving person from watching the 2-d projected images over time. A dancer could repeat the motions depicted in the film.*" [96]. Loy *et al.* [97] reconstruct the 3D body configuration in tennis scenes by first detecting the key poses of the sequence, measuring shape similarities defined by the contours and selecting the poses. Loy requires user interaction for labelling the locations of the joint in the key frames, while the joints in the non-key frames positions are interpolated and aligned by edge information. 2D to 3D lifting uses again an operator for setting the 3D positions of joints in the key frames. The method is labour intensive, requiring considerable user interaction and a complete sequence to identify the key frames. Therefore it is not tractable for on-line motion recognition or automated scene analysis.

Currently there is great interest in pose detection from monocular images, since the multi-camera hardware is not available for low cost or restricted view applications. Lacking direct recovery of 3D geometry implies the use of a-priori information. Howe [98] proposes mapping from silhouettes to poses by using a lookup table with a turning angle metric and the Chamfer distance of the silhouette. Agarwal and Triggs [99] reconstruct the body pose of an articulated model, learning silhouette to image correspondences. The silhouettes are modelled by shape context distributions, and are searched for within the 55-dimensional pose space using regression. The detection is extended to tracking by means of a particle filter (see section 2.2.2). Chiu *et al.* [100] reconstruct 3D poses using iterative look-up methods in a posture library. As it assumes orthographic projections and labelled input, the method is limited, however physical constraints such as body part length ratios, joint limits, pivotal posture reference from the posture database and feet-floor contact points are used as prior information for tracking.

İkizler and Forsyth [32] use learnt motion capture data to estimate the pose of the limbs, but not directly of the whole body. This allows a relatively small set of motion data to recover a wide range of poses. Although Lv and Nevatia [41] do not recover 3D pose, but perform action recognition directly, their silhouette shape context descriptor search with Pyramid Match Kernel and is also promising for 3D pose recovery. Shakhnarovich *et*

*al.* [101] use Locality-Sensitive and Parameter-Sensitive hashing on an edge map, that is a set of filter responses. The search over the training dataset for the $k$-nearest neighbours of the features results in poses for which the mean is the pose estimate. Similarly, Poppe [102] uses oriented histograms, to scan the whole dataset and to estimate the best pose from the weighted mean of the nearest neighbours evaluated with Manhattan, Euclidean, cosine and $\chi^2$ distance metrics.

### 2.2.2 Stochastic tracking

Stochastic tracking methods do not provide an exact and single estimate, but maintain a p.d.f. over time, aiming to search only in the areas where the solution is likely to be found, therefore saving resources where a solution is unlikely.

#### Kalman filters

The family of *Kalman filters* (KF) [61, 103, 104] estimates a system of real, frequently hidden, parameters in a noisy environment by making indirect measurement.

The basic KF is constrained to linear systems. However, the *Extended Kalman Filter* (EKF) overcomes this, by piecewise linearisation of the system and the observation models.

The *Unscented Kalman filter* (UKF) [103], is based on the unscented transformation introduced by Julier and Uhlmann, and uses the ingenious observation that it is easier to approximate a Gaussian distribution than it is to approximate arbitrary nonlinear functions. The unscented transformation computes the result of a distribution propagated through a non-linear function. The initial distribution is represented with a set of deterministically chosen sample points that completely capture the true mean and covariance of the state variable.

The KF, EKF and UKF have been extensively applied to blob tracking applications, but the interest here is in the more complex problem of tracking for interpretation of human movement. The demand for probabilistic human tracking was recognised early. Bregler [24] uses a Kalman Filter for tracking and location of independent blobs, where measurements result from detection of the same motion (*i.e.* image gradient) and colour regions, integrated later with EM into humans. Baumberg and Hogg [92] use a Kalman filter to determine active shape parameters as well as the global position. Caporossi *et al.* [105] use first order Kalman filter tracking position and velocity of blobs, while Wren

*et al.* [106] estimate with a KF the 2D human multi-blob human.

The system of Zhao *et al.* [107] is a 2D articulated human tracker, with trunk, head and the two legs modelled with a rectangle. Combining KF for consistent prediction and tracking with the mean shift implemented on each body part blob, the parts are assembled with MAP criteria, achieving fast (20Hz) articulated human tracking.

To conclude, KF are fast and accurate tracking filters, but they work only with linear or linearisable models and Gaussian parameter distributions. The observations require detectors that have a well known definition.

### Particle filters

The Kalman filter fails to track highly non-linear systems with non-Gaussian parameters. Since human motion is both non-linear and non-Gaussian, an alternative method is required. The *Particle Filter* (PF) is a plausible candidate since it handles both these requirements, and recent technological advances provide the processing power required by the explicit representation of the parameter distributions.

For comparison, the most common forms of the Kalman and Particle filters are presented in table 2.4. All KFs assume Gaussian state variables and, except the UKF, a linearisable dynamic model of the system. This is particularly inconvenient for multiple model systems with model switching required for complex systems.

The two key elements of the PF are the motion update, defined by prior distribution, and the observation likelihood. For better quality, several tracking algorithms combine the PF with other methods to integrate stronger prior knowledge into the PF proposal generation or to enhance the measurements.

A number of authors have taken the basic PF and combined it with other techniques to produce hybrid operators with more desirable properties, at least that was the intention. Shan *et al.* [108] combine particle filter and mean shift into the *mean shift Embedded Particle Filter* (MSEPF). With a mean-shift step, the number of particles is reduced. Since hands and the head have a well-defined skin-colour model, the mean shift provides strong measurements. Combining the two different search classes, stochastic and model-driven (particle filter) with deterministic and data-driven (*e.g.* mean shift) results in robust tracking in spite of occlusion and near similar regions. However, it is uncertain how the MSEPF performs if the object becomes less similar to the target model during tracking. Further, the method is relatively costly for tracking only a 2D single blob. Similarly,

| Filter | Dynamic model | State variables | Applicability | Computational complexity |
|---|---|---|---|---|
| Kalman | Linear | Gaussian, Mean and Covariance | Linear systems only | Simple |
| Linearised Kalman | Linearised | Gaussian, Mean and Covariance | Nonlinearity up to $1^{st}$ order | Tracking is computationally cheap, but needs complex precomputation for nominal trajectory |
| Extended Kalman (EKF) | Linearised | Gaussian, Mean and Covariance | Nonlinearity up to $1^{st}$ order | Complex, needs the Jacobian for linearisation |
| Unscented Kalman (UKF) | Nonlinear | Gaussian, Samples points | Nonlinearity up to $2^{nd}$ order | Simple, may increase with number of samples |
| Particle filters | Nonlinear | Any distribution | Result's distribution should be similar to the initial distribution | Basic algorithm is simple, complexity increases with the number of particles; complexity highly depends on likelihood complexity evaluated for each particle |
| Extended Kalman Particle Filter (EKPF) | Nonlinear | Any distribution | Nonlinearities are better modelled than with PF (with less particles) | More complex, in the update phase the particle uses EKF |
| Unscented Particle filter (UPF) | Nonlinear | Any distribution | Nonlinearities are better modelled than with PF (with less particles) | UKF is used for particle estimate, cheap as basic PF |

Table 2.4: Stochastic filter comparison chart.

Maggio and Cavallaro [109] combine PF with mean shift in a hybrid tracker for better proposal distribution generation. The integration of the PF with Adaboost detection, presented by Okuma *et al.* [110], applies the linear combination of the Adaboost detection probability and the next state transition priors, generating a strong proposal distribution and enhancing the PF for multi-modal posteriors. Adaboost also allows detection and initialisation of new objects.

Saboune and Charpillet [111] use a semi-deterministic particle filter, the Interval Particle Filter. The grid defines the ordered set of the possible discrete values of the selected model parameters with an expected change. At each iteration each of the $n$ particles is spawned by this grid into the $k$ neighbours of the original particle, multiplying by $k$ times the particle set. Each new particle is evaluated and the highest $n$ are kept to model the new distribution. No dynamic model is used, and therefore, depending only on the grid domain, it adapts to unseen motion. However the grid neighbourhood dimensionality increases exponentially with the number of the parameters. Saboune and Charpillet remove this disadvantage in part, adjusting the 3D global parameters first, then applying IPF only on the limb parameters, generating for each particle *only* $k = 81$ neighbours. Similarly, Lanz [112] uses a Hybrid Joint-Separable model for fast, blob-like multi body tracking. Thayananthan *et al.* [113] use grid based tracking, with hierarchical partitioning of the state space and a tree based iterative approximation of the posterior, with repetitive hierarchical detection.

Taycher *et al.* [114] use grid filters with model parameters sampled into piecewise constant domains that convert the continuous state space into finite discrete values. The constant random field replaces the usual PF measurement phase, and uses the learnt observation potential for quick computation of the model-image joint probability. Sminchisescu and Triggs [115] modify the PF considering the kinematic structure and a motion model. A symmetrical body part flips resulting in the same monocular observation. They also add posterior pruning by removing low probability predictions. The method is accurate for tracking a short (4s, 100 frames) sequence of agile movement.

Apart from the posterior generation, defined by the system dynamic model, the many different methods of likelihood computation can be generated within the context of a PF. In general, direct likelihood evaluation is not possible, since it requires a good, fully parametrised body part detector. Therefore, PFs use *analysis by synthesis*, or the *generative approach*: the model is projected into the image, and the resulted projection is directly compared with the image. How well the image matches the projection is the likelihood of the projection, and hence of the projected particle. Generative models act against the discriminative tracking methods used in KF, Mean shift, *etc.* that require extracted image features and the construction of the tracked model from these features. Wachter and Nagel [116] remark in the context of a modified KF that direct model fitting to the image data, compared to feature-based methods, has the advantage of having no heuristics

for feature extraction nor needs a measure to compare these features. The avoidance of feature extraction is a great advantage; however it introduces complex and multiple likelihood computation. Even so, this is more tractable than incorporating 2D measurement for 3D or inverse kinematic modelling.

*Partitioned Particle Filters* (PPF) were introduced by MacCormick and Isard [128]. Unlike the basic PF, the PPF divides the high dimensional parameters into sets evaluated and updated independently (figure 2.3).

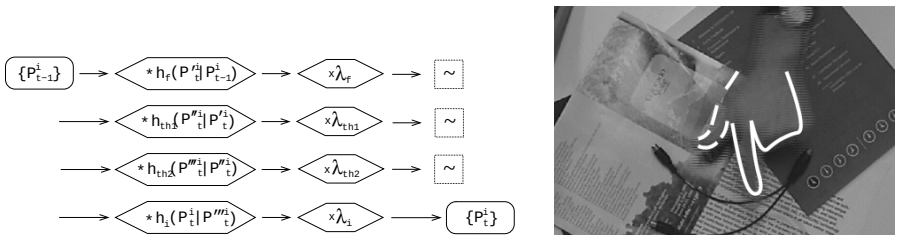


Figure 2.3: Partitioned particle filter [128]. Parameter distribution generation (left) at time $t$ from distribution at $t-1$ by dynamic model update ($*$), observation likelihood integration ($\times$) and resampling ($\sim$). Example of the tracked hand model (right).

The parameter space is divided into several partitions. For a tracked, articulated human hand, seven parameters from the total, are divided into four partitions $\phi = \phi_f \times \phi_{th1} \times \phi_{th2} \times \phi_i$: the *first* ($x = f$) finger, the *thumb base* ($x = th1$), the *thumb tip* ($x = th2$) and the *index finger* ($x = i$), with 4 respectively $3 \times 1$ parameters for each of the partitions.

The $h_x$ dynamics and likelihood $\lambda_x$ of the parameter partitions are independent, after the *first* set is fixed, because of the physical independence of the fingers. The particles are sequentially updated in 4 integrations, each being similarly to a single resampling iteration, but with different $q_x$ and $\lambda_x$ for each $x$.

Hence, MacCormick and Isard [128] partitioned the parameter space, tracking parameters independently with independent dynamics and likelihood functions. Partitions that are error prone, or with parameter errors (such as global parameters) that propagate to other partitions, contain more particles, while less important partitions have reduced parameter numbers and result consequently in faster processing. They argue that partitioned sampling is equivalent to a hierarchical search. Their partitioning is valid only if the parameters are independent or loosely independent. However, when hierarchic dependence is present, such as for human body parts, then simple partitioning is not optimal.

Deutscher *et al.* [117] used an *Annealed Particle Filter* (APF) with dynamic hierarchical partitioning, however multiple modes in the parameter space make it inappropriate to

| Authors | Monocular | Tracked target | Tracked parameters | Number of particles | PF variant |
|---|---|---|---|---|---|
| Deutscher *et al.* [117] | N | AB | 34 | $100 \times 10$ layers | Annealing, adaptive diffusion and crossing over |
| Lee *et al.* [118, 119] | N | AB | 23/32 | 100 | Head, hands, torso |
| Green and Guan [26, 27] | N | AB | 43 | NA | Equilibrium and physical limits added |
| Sidenbladh *et al.* [120] | Y | AB | 12 | 1000 | PCA compressed constant velocity model |
| Sidenbladh *et al.* [121, 122] | Y | AB | 25 | NA | Limb appearance likelihood |
| Sminchisescu and Triggs [115, 123, 124] | Y | AB | 30 | NA | Joint angle limit and body non-self-intersection constraints; Covariance sampling |
| Kim *et al.* [125] | Y | AB | 40 | 100 | NA |
| Ning *et al.* [126] | Y | AB | 12 | 300 | Motion synthesis |
| Saboune and Charpillet [111] | Y | AB | 4 | 81–6561 | Motion estimation |
| Pantrigo *et al.* [127] | Y | Upper body | 18 | $\geq 2500$ | Heuristics: path relinking and scatter search |
| MacCormick and Isard [128] | Y | Hand | 7 | 990[a] | Partitioned sampling |
| Shan *et al.* [108] | Y | Hand | 4 | 20 | Mean shift embedded PF (MSEPF) |
| Zhao *et al.* [129] | Y | Arms | NA | 512 | Appearance likelihoods |
| MacCormick and Blake [130] | Y | Head | 8–9 | 750 | Partitioned sampling |
| Kang and Kim [131] | Y | 2D contour | 40 | 100 | Weight suppressed around areas of other tracker |
| Zhang *et al.* [132] | Y | Contour | NA | NA | Annealing and MCMC |
| Maggio and Cavallaro [109] | Y | BB | 3 | 30 | Mean shift |
| Okuma *et al.* [110] | Y | BB | NA | NA | Adaboost |
| Lanz [112] | Y | BB | 4D[b] | 100–500 | Hybrid Joint-Separable model |
| Yang *et al.* [133] | Y | BB | 4D[b] | 1000 | NA |
| Osawa *et al.* [134] | N | BB | 4 | 300 | Environment model |

Table 2.5: PF based human tracking. Glossary: AB - articulated full body, BB - blob full body, NA - information not available.

[a] summed particle number for partitions
[b] per object

use unimodal co-variances to define the partitions. Unlike the PPF, the APF has several iterations, but these are not performed on partitions of the parameter set. Instead, the likelihood function is changed: the initial iterations have a flattened likelihood, while the latter ones are more peaked. As a result, the first iterations overcome local likelihood minima, while final likelihoods assure a good fit. Iterations are called layers of the APF (figure 2.4).

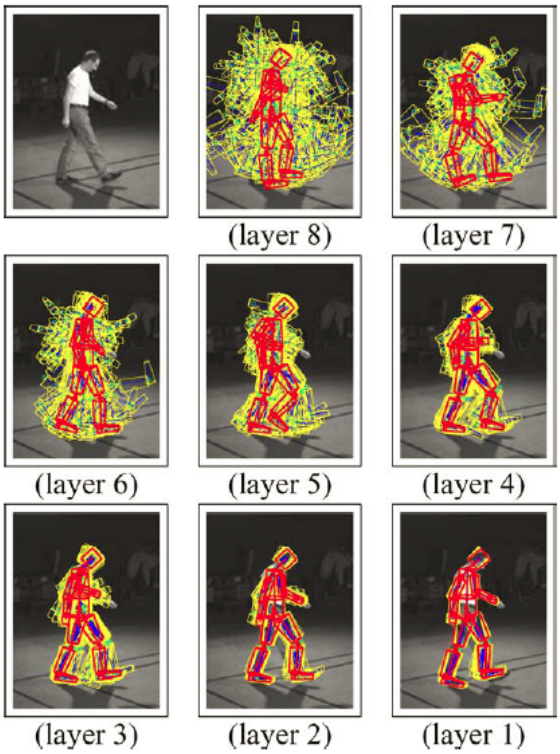



Figure 2.4: APF convergence over multiple layers [117].

The second enhancement of Deutscher *et al.* is a hierarchical search. This automatically updates the motion model, on the first levels of which parameters have a large variance, but this is reduced on later levels. For this, the covariance is computed proportional to the covariance of the current particle set on each level.

Third, Deutcher uses a crossover operator, inspired by genetic algorithms, that mixes parameters of two particles. Unfortunately it can be expected that, if the parameters are not ordered using a rule that related parameters are close, and due to the hierarchical dependence of the particles, the crossover operation incorrectly mixes unrelated parameters.

According to Deutscher, the APF with 10 annealing layers effectively searches the hierarchical parameter space of a human observed with well defined silhouettes by 3 cameras

in 4–5 seconds long sequences. The reported processing speed is slow, 15 secs/frame on a single processor 1 GHz PIII Linux box.

Table 2.5 provides an overview of the PF methods used for human tracking. These systems either track the whole body as a blob, or track some sub-set of parts of that body, or use a full, articulated model. They are based either on monocular or stereo images as input.

Maintaining a large set of particles is costly, but necessary. Otherwise, it is subject to a curse of dimensionality, the poor representation with a limited number of particles of the underlying complex probability distribution. That is highlighted by high dimensional models with more than 18–25 parameters (table 2.5). For faster processing, modified versions of the basic particle filter are aiming to reduce the number of particles.

Since PFs are non-deterministic, there is no exact reproducibility and different runs on the same data can result in different outcomes.

| Pros | Cons |
|---|---|
| Non Gaussian parameter representation | |
| Multiple hypotheses | Non deterministic |
| Recovery from false observations | Computationally costly |
| Non linear system model | Multiple objects |
| A-priori information integration | |
| Multiple and different observations | |
| Multiple objects | |

Table 2.6: Advantages and disadvantages of PFs.

The advantages and disadvantages of the PFs summarised in table 2.6 result from the explicit representation of the tracked distribution by samples. Thus, the model configuration does not have to be Gaussian, and multiple high probability hypotheses may coexist, represented by subsets of particles. This allows recovery from errors or false estimates resulting from erroneous or missing input. The system model need not be linear, since the proposal generation resolves the model dynamics and can also include other a priori known constraints. Multiple objects can be modelled by priors for reasoning on their reciprocal occlusion, coherent movements and interactions. However, multiple object will multiply the tracked parameter dimensionality.

**Nonparametric belief propagation**

Sudderth *et al.* [135] define *Nonparametric Belief Propagation* (NBP) as an extension of particle filtering for the more general vision problems that graphical models can describe. NBP is a probabilistic method, similar to PF, but adapted for tracking multiple targets with a strong structural link (*e.g.* an articulated hand [136] or articulated human body [137], figure 2.5). The NBP has a graphical model, a graph structure, through which the distribution of the parameters, as messages, are propagated between the neighbouring nodes of the graph. The distributions are propagated between neighbouring targets or parts by passing local, probabilistic messages, represented by particles. The messages, at each node, are generated with the Gibbs sampler, followed by a Monte Carlo approximation of the outgoing message. NBP is similar to a pairwise Markov random field [136].

When tracking articulated hands for example [136], the parameter space dimensionality is multiplied from 26 to 96, however by introducing the priors of the kinematic relations, Sudderth *et al.* reduce the effect of the dimensionality increase with the separated likelihoods of the different parts. However, the complexity of the message passing, involving several iterations, means that tracking is very slow (4 minutes/frame, platform not reported).

*Particle Message Passing* (PAMPAS) [138] is an equivalent formulation of NBP. PAMPAS works with continuous variable distributions, by introducing a Particle Filter over Belief Propagation. The 43-dimensional jointed object detection tests show good matches between models and data, however the task allows a straightforward likelihood formulation on a binary image and without occlusion handling.

Sigal *et al.* [137] argue that NBP is better than Deutcher's Annealed PF [139], which loses the target quickly, without the possibility of recovery. Edge and silhouettes are used as image likelihoods, while importance function is computed by part detectors using multi-view Eigenspace analysis.

### 2.2.3 Conclusions

While the Lucas and Kanade feature tracker is an appearance based method processing the intensity image, the mean shift uses a histogram (a p.d.f.) of the tracked region. The first method is valuable when only small changes occur in appearance of the object. However, the second can deal with 2D or 3D rotation or translation of the object as it learns a

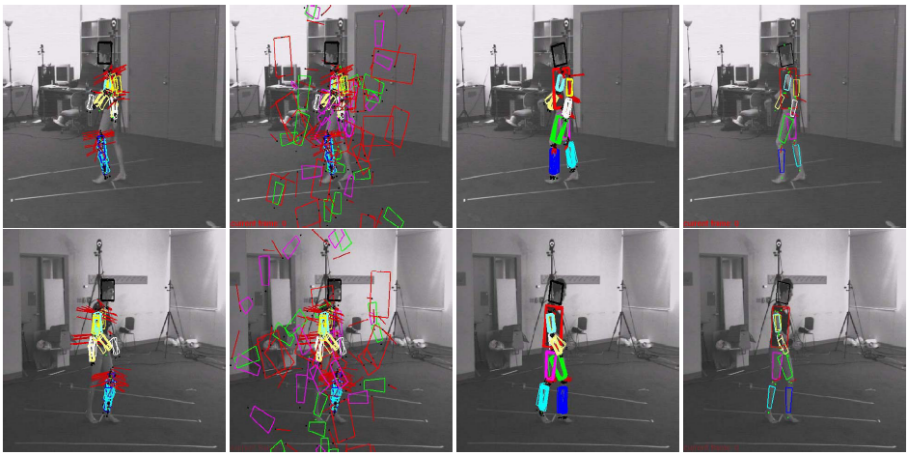


Figure 2.5: Nonparametric belief propagation [137]. Body parts are initially detected over the whole image in the two camera views shown, however they converge to form the human body.

density of features that usually are less affected by transformations or smaller occlusions. Snakes allow fast processing, however are limited to the 2D and to the contours they are trained on. 3D structure from 2D lookup techniques are promising, they are generally fast and need a single view only, are independent of the body model representation, if the body model changes, they require only retraining. However, for generality they require extensive training from multiple views, they depend completely on the training data, and they cannot recognise untrained poses. Further, the ability to generalise across different subjects is not proven. In conclusion, the current trend is from deterministic tracking towards stochastic techniques, frequently implemented by particle filters, with the advantages of modelling non-Gaussian, multi-modal state spaces and dynamics and allowing multiple hypothesis and probabilistic evaluation of the results. However it is resource demanding compared to deterministic and Kalman filters, especially when applied in high dimensional parameter spaces.

## 2.3 The use of prior knowledge in tracking

Prior knowledge is all the information the implementer knows or collects *in advance* in the design and learning phase of the algorithm, and before the actual data is processed. In this section, we present the way in which 2D and 3D human models are represented for tracking, and their dynamics, as well as the scene model, defining the relation between the three dimensional world and the two dimensional image.

Blind parameter optimisation is computationally poor, especially if the search space of parameters is combinatorially large. This leads to the need for prior knowledge that directs the search, and restricts it to the possible range, or a likely one. Carefully designed priors are important since, if too strong, they exclude the solution, or if too weak, the search space complexity remains intractable.

Bayes' rule

$$\mathcal{P}(\mathrm{x}|\mathrm{O}) = \frac{\mathcal{P}(\mathrm{O}|\mathrm{x}) \cdot \mathcal{P}(\mathrm{x})}{\mathcal{P}(\mathrm{O})}, \tag{2.2}$$

gives insight into incorporating priors into the conditional probability of $\mathcal{P}(\mathrm{x}|\mathrm{O})$, the current parameter x conditioned by the observation O. The likelihood, $\mathcal{P}(\mathrm{O}|\mathrm{x})$, is the conditional probability of observing O given x. For clarity, further in this work $\lambda(\mathrm{O}|\mathrm{x}) = \mathcal{P}(\mathrm{O}|\mathrm{x})$ will denote the likelihood. $\mathcal{P}(\mathrm{x})$ and $\mathcal{P}(\mathrm{O})$ are the priors precomputed by training or other means (*i.e.* assumed uniform). $\mathcal{P}(\mathrm{O})$ from equation (2.2) is independent of x, therefore can be factored out without evaluation while maximising the current parameter x.

Priors categorised on their dynamics are

- *global/static* priors ($\mathcal{P}(\mathrm{x})$) being independent of their preceding state, global priors include articulated models and scene configuration,
- and *local/dynamic* priors ($\mathcal{P}(\mathrm{x}_t|\mathrm{x}_{t-1})$ defined by previous configurations. These provide the expected change of the parameters, such as the dynamics of the model, the current model (for switching models), scene dynamics.

Priors incorporate information about the tracked target, the physical medium and the sensory system (table 2.7). Therefore 2D and 3D human representations and dynamics are reviewed firstly, followed by consideration of dynamic and scene models. Finally, behavioural priors as yet ignored by vision systems are summarised.

### 2.3.1 Human body models

Human tracking implicitly assumes that the target is a human, though the model detail varies from representation to representation including 1D and 2D curves, 2D blobs, and 2D and 3D articulated objects. This section considers a simple classification based on 2D and 3D approaches.

| Author | Priors | Measurements | Methodology | Applications |
|---|---|---|---|---|
| Agarwal and Triggs [99] | Learnt Pose to Silhouette mapping, Dynamic mode | Motion smoothness (for regression) and Distance from regression (for PF) | PF with Pose recovery | Monocular articulated human tracking |
| Antonini *et al.* [140] | Ground plane, camera calibration discrete choice model | Correlation | Correlation maximisation, with post-filtering | Tracking multiple persons in occluding 3D space |
| Böhme [66] | Head contour, skin colour, face model | Detectors for head, skin and face | PF | Customer assistant robot |
| Bălan *et al.* [141] | Calibrated camera, Stationary backgrounds, Single subject, self-occlusion only, 0GM and 1GM model | Edge & Silhouette | PF | NA |
| Baumberg and Hogg [92] | Pedestrian 2D active shape | Edge response along the contour normal | Kalman Filter | Pedestrian tracking |
| Bobick *et al.* [142] | Ground plane, current context (story), exit/enter areas | Colour, velocity and size | Detection | HCI / Virtual environment |
| Bregler [24] | Constant velocity model | Motion and colour coherence | Kalman Filter | Action recognition with HMMs |
| Caporossi *et al.* [105] | Colour model, Motion linearity | Silhouette(foreground) and colour | Kalman filter | Head pose estimation, Agent tracking |
| Cheng and Chen [65] | Colour consistency, largest blob | Colour of Moving areas | Detection | Multiple people tracking |
| Cremers *et al.* [143–145] | Dynamical shape model | Colour consistency | Level-Set | 2D tracking |
| Delamarre and Faugeras [77] | 3D human model, Camera calibration | Forces on Edges/Contour similarities | Detection | Articulated human tracking |
| Deutscher *et al.* [117, 139] | 3D model, Calibration | Edge and silhouette | PF with annealing | Motion capture |
| Fuentes [53, 54] | Motion contiguity,size, aspect ratio | Foreground | Detection | Unattended luggage, Fall, Hiding, Vandalism, Fight detection |
| Green and Guan [26, 27] | Calibration | Edge and silhouette | PF | Multiple human tracking |
| Haritaoglu *et al.* [52] | 1GM motion, 2D posture database | Silhouettes, Texture | Add-hoc, detection | Multiple human tracking, and simple behaviour analysis |
| Howe [98] | Silhouette - 3D configuration mapping | Shape (Silhouette matching by Chamfer distance) | Detection, Markov chaining, smoothing | Single view pose reconstruction |
| Kang and Kim [131] | 2D learnt shape | Moving edge direction | PF | Multi-human tracking |
| Kim *et al.* [125] | Constant velocity model | Edge distance (Chamfer) | PF | - |
| Krahnstoever *et al.* [146] | Ground plane, Calibrated cameras | Colour histogram, Part based | Multi camera detection | Abandoned bag detection |
| Krumm *et al.* [17] | Calibration, Ground Plane, People shape of blobs, Entry/Exit zone | Colour histogram | Detection | Multiple human tracking |
| Lanz [112] | Ground plane, Calibrated cameras (human sizes) | Global colour histogram with part based, occlusion reasoning | PF | Multiple human tracking |
| Lee *et al.* [118, 119] | Camera, 3D model | Silhouette edge and region, Body part detectors | Motion capture | Head, hands, torso |
| Lee and Elgammal [147] | View dependents silhouette trajectories | Silhouette shape matching | PF | Single view pose reconstruction |
| MacCormick and Blake [130] | Head shape, Parameter inter-dependence | Edge | Partitioned PF | General tracking |
| MacCormick and Isard [128] | Hand model with separable parameters | Edge | Partitioned PF | HCI |
| Maggio and Cavallaro [109] | Zero order adaptive motion | Bhattacharyya colour similarity | PF-Mean shift hybrid | Blob (human, car, *etc.*) tracking |
| Nickel [50] | Posture probability, Posture dynamics | Skin colour and motion | Multi hypothesis tree | Pointing gestures |
| Ning *et al.* [126] | Learnt periodic motion, Camera projection | Edge and silhouette | PF | Motion synthesis |
| Okuma *et al.* [110] | Motion model (AR), Adaboost detection | HSV colour | PF | Multi object tracking |
| Osawa *et al.* [134] | Ground plane, Constant velocity model | Silhouette (BS), Height from the ground plane | PF | Tracking multiple persons in occluding 3D space |
| Pantrigo *et al.* [127] | Upper body 2D model | Canny Edges | PF with evolutionary, local optimisations | Planar articulated motion |
| Porikli and Tuzzel [88] | Shadow model | Colour and Silhouette (foreground) | Mean shift | Face and hand tracking |
| Remagnino *et al.* [93] | Shape priors, 3D models, Calibration, ground plane | Head and region detector | Kalman filter | Pedestrian and car surveillance |
| Saboune and Charpillet [111] | Calibration | Silhouette | Interval PF | Motion estimation |
| Seeman *et al.* [72] | Implicit shape model codebook | Shape Context Descriptors | Detection | Pedestrian tracking |
| Shan [108] | 1GM motion | Skin colour and motion | Mean shift embedded PF | Wheelchair visual hand control |
| Sidenbladh *et al.* [120, 121] | 3D model, $1^{st}$ motion model | Limb texture | PF | General articulated tracking |
| Sidenbladh *et al.* [122] | 3D model, Implicit motion model | NA | PF | General articulated tracking |
| Siebel *et al.* [94] | Pedestrian 2D active shape | Edge and silhouette, Foreground | Detection | Gait and action recognition |
| Sigal *et al.* [137] | Body articulation, learnt limb relations | Texture (multi scale edge and ridge) | NBP | General articulated tracking |
| Sminchisescu and Triggs [115, 123, 124] | Joint limits, Model proportions, Collision avoidance, Camera calibration | Edges and Silhouettes | PF with Kinematic Jump Process | Articulated human tracking |
| Stenger *et al.* [148] | 0GM model | Edge based Chamfer distance and skin colour foreground/background ratio | Repeated detection | HCI domain |
| Sudderth *et al.* [136] | 0GM model | Edge: both Chamfer distance and edge orientation, Foreground and Background Colour model ratio | NBP | Hand tracking |
| Taycher *et al.* [114] | Parameter-image joint distribution modelled by CRF | Texture(edge direction histogram) | Conditional Random Field | Single view pose recovery |
| Viola *et al.* [74] | Pedestrian model texture and motion | NA | Detection | Pedestrian tracking |
| Wren *et al.* [106] | 1GM motion | Skin colour | Kalman filter | HCI |
| Wu and Yu [149] | Trained two-two-layer random field model of the shape and observation | Edge map | PF | Pedestrian tracking |
| Yang *et al.* [133] | 0GM motion | RGB colour and edge orientation histogram | PF | Fast single and multi-human blob tracking |
| Zhang *et al.* [132] | 2D human shape | Edge, silhouette, skin colour, region similarity | PF | Articulated human tracking |
| Zhao *et al.* [129] | Linearised Switching Models | Edge and texture (colour) | PF | Action recognition |
| Zhao *et al.* [107] | Human structure, Colour consistency | Colour | Mean shift, Kalman filter, multiple hypothesis | Fast multi person tracking in low resolution images |

Table 2.7: Survey of the current algorithm priors, measurements and methodology. The last column shows the actual or possible application domain.

**Two-dimensional human models**

Wang *et al.* [150] represent humans in one dimensional space with the distance of the unwrapped silhouette contour pixel from the silhouette centre. This is extremely sensitive to the localisation of the silhouette centre, affected by the background subtraction accuracy. Although they provide over 80% recognition for gait classification with spatial-temporal correlation analysis, 1D models are valueless in tracking since they require the human blob to be found first, which assumes previous 2D modelling.

Therefore, the simplest from of human modelling in tracking is through *2D position* and *rectangular windows* [54, 110, 112, 133]. The first advantage is a reduced number of parameters, allowing straightforward processing for simple behaviour with 2D features (*i.e.* position, speed, trajectory). Unfortunately, oversimplifying the data results in behavioural analysis being limited to global activity and simple interaction only.

Elgammal and Davis [151] use *elliptical blobs* to model humans. Blobs are represented by the independent density functions of the colour, the vertical and the horizontal density. The probability of a model is defined using these densities relative to the origin position. Elliptical models [109, 140] approximate better the human shape when compared to bounding boxes, but still provide reduced information about human dynamics.

Moving towards detailed human models, very strong prior assumptions on the shape of the human silhouette allow Haritaoglu *et al.* [52] to detect limbs and joints as projection extrema, as high curvature points of the silhouette contour, and as convex hull vertexes around the silhouette. Matching against a 2D posture database, with 4 main poses each and 3 views results in an actual posture with joint positions on the silhouette. Baumberg and Hogg [92] use *deformable contour* or *active shape* models for human tracking. Their deformable model, learnt by training from a silhouette database, consists of a closed B-spline curve controlled by a set of control points. To speed up the tracking process, the parameter space dimension is reduced to 14 parameters by PCA. The obtained model performs well in real time tracking. Similarly, Remagnino *et al.* [93], Siebel [94] and Kang and Kim [131] use learnt 2D active shape models for pedestrians, but B-splines are also used for head [130] and hand [128] tracking. These deformable contour models have details of the lower limbs, but since the the upper limb configuration has much greater variety of movement, these are not modelled. For this reason, and because of the single plane, 3D model reconstruction from 2D is not possible.

Zhang *et al.* [132] use a multi view 2D model, with learnt shape priors represented as landmark points approximating the contour of the silhouette. Hierarchical composition of the landmark points into view and body parts makes the model dynamics simple, but it is limited to trained poses.

Leventon *et al.* [152] employed a deformable contour representation for segmentation, with prior information computed from an assumed Gaussian probability distribution computed from training shapes. The process defined the initial curve as the zero level set function of a higher dimensional surface. This surface evolves to converge on the boundary of the object of interest, using a maximum a posteriori (MAP) estimate.

*Articulated models* [127, 129, 132, 153–155] characterise each body part independently along with the body position. Wren *et al.* [106] use an articulated blob model, but assume a single person in the image. The human is built in a bottom-up manner, first individual blobs are detected, then head and hands, and feet locations are identified with strong colour priors, and blobs are assembled by contours and colour consistency into humans.

Huttenlocher *et al.* used pictorial structures [153–155] to achieve flexible model tracking of 3D articulated objects. They define the pictorial structure as a parametric spatial model composed of multiple parts with loose spring connections between certain pairs of points. Spatial relations between the parts are learnt. Each part of the pictorial structure is represented by a simple appearance model with a deformable configuration. With dynamic programming the global minimum is found by a tree search. This model is able to handle changes in viewpoint and allows self-occlusion in face feature, car and human motion tracking.

The reviewed methods are effective for specific applications where task is well defined and fixed, and camera position is constant after the initial setup and training. However, to achieve generality, independence from viewing, and easy adaptability requires 3D modelling.

#### Three-dimensional models

*3D blob models* [134, 156, 157] represent humans as volumes defined by their ground plane positions, heights and widths. This allows contact and occlusion reasoning, and metric distance computation. They are generally more valid and independent of the viewpoint or application [116]. Since they are not detailed, they do not include information about the orientation and position of limbs for example, and therefore simple articulated models

capturing physical appearance are preferred for behavioural reasoning.

On the other hand, *high detail 3D models*, such as SCAPE [158] aim for accuracy. However tracking them is ineffective, though Balan *et al.* [159] use a cylindrical model refined deterministically into the detailed model (figure 2.6). A computationally simpler 3D humanoid model of Hilton *et al.* [160, 161], used for computer animation rather than behaviour recognition, allows fast model recovery at the cost of a multi-camera system.

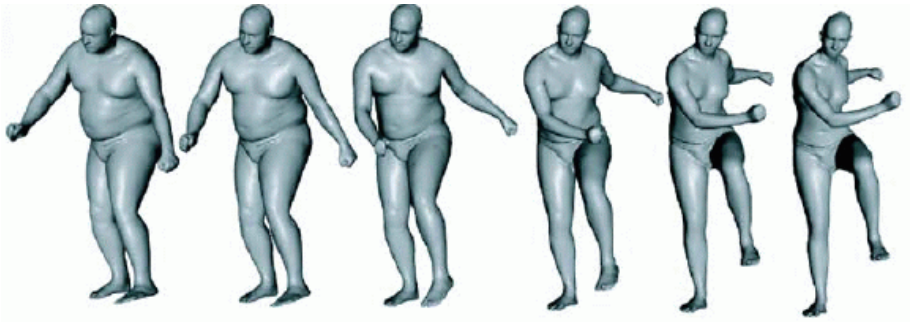


Figure 2.6: Instances of the SCAPE model [158], while tracking and adjusting the body parameters.

*3D Articulated human models* differ from author to author (table 2.8) in how many individual body parts are included, how these body parts are represented, how many *Degrees Of Freedom* (DOF) are represented in the body and from these how many are independent, are recovered during the tracking or considered constant after fitting to an initial body. They define the parameter range according the physical limit of the body [26, 116], or define inter-connectivity rules [137].

A human body is articulated by its skeleton, but what is observed has much more bulk and must be somehow represented on top of the skeleton by some kind of geometric primitives. Green and Guan [26, 27] use a clone-body-model. Each body part is made of a rigid spine with pixels radiating out with nine parameters: the 3D coordinates, the colour, the accuracy of colour and radius, and elasticity. Body parts are dynamically sized and texture mapped to real body parts. The texture allows larger variations in somatotype, gender, age; greater accuracy with exact sizing of clone-body-parts; increased region tracking accuracy; region patterns such as ear, elbow, knee assist in fixing orientation. The body parts used are the head, clavicles, trunk, upper arms, forearms, hands, thighs, calves and feet. To model the human body parts, Plänkers and Fua [162] use articulated soft objects, called metaballs. Metaballs are ellipsoids attached to skeleton segments, and 230 metaballs model the full body. Parameters are adjusted by the silhouettes and 3D

| Author | Body part | Body part model | DOF | Tracked parameters |
|---|---|---|---|---|
| Saboune and Charpillet [111] | 17 | rigid segments | 31 | 4 |
| Sidenbladh *et al.* [120] | 9+3 | cylinders & sphere | 25 | 12 |
| Ning *et al.* [126] | 13+1 | conical frustums & sphere | 40 | 12 |
| Delamarre and Faugeras [77] | 27 | conical frustums, spheres, and right parallelepipeds | 20 | 20 |
| Lee *et al.* [118, 119] | 11 | conical frustum | 23/32 | 23/32 |
| Sidenbladh *et al.* [121] | 10 | cylinders | 25 | 25 |
| Deutscher *et al.* [117] | 15 | elliptical cross-section frustums | 29 | 29 |
| Plänkers and Fua [162] | 230 | ellipsoids | 27[1] | 27 |
| Sminchisescu and Triggs [123, 124] | 16 | super-quadric ellipsoid | 30 | 30 |
| Kim *et al.* [125] | 17 | conical frustums | 40 | 40 |
| Wachter and Nagel [116] | 15 | elliptical cross-section frustums | 23 | 46 |
| Sigal *et al.* [137] | 10 | conical frustums | 60 | 60 |
| Green and Guan [26, 27] | 16 | 3d surface with spine axis | 43 | 129 |

Table 2.8: Three-dimensional human models differ in the number of elementary geometric shapes used, the type of these shapes, the total DOF of the model, and how many of these parameters are adjusted during tracking, direct parameters being possibly compressed or augmented with other dynamic parameters.

observations from two cameras. Body shape and position are controlled by a state vector, which for the upper body has 27 DOF. The Sminchisescu [124] model has 30 joint variable parameters, plus eight fixed internal proportions, and nine deformable shape parameters for each super-quadric ellipsoid body part with discretized 3D surface meshes.

Other models have emerged from commercial 3D motion caption formats. For example, Ong *et al.* [163] use the Biovision format with Euler angles for each joint and a translational offset for the entire body. The CMU Mocap database is presented in Acclaim ASF/AMC format and Coordinate 3D (C3D). Sigal *et al.* [137] uses 10 tapered cylinders each with five fixed and six estimated parameters. This model is similar to the HumanEva model, body cylinders being arranged hierarchically, and specified relative to their parents with

a generic 6D transformation. Similarly, Chiu *et al.* [100] use the hierarchical structure of the MPEG-4 Body Definition Standard.

### Common modelling problems

The following problems are common to all 2D and 3D, blob and articulated models. First, tracking and behavioural analyses assume the previous detection of the tracked object, which is not trivial. *Detection* and *initialisation* is generally simple and ad-hoc, and based on motion [88, 94, 150, 164], skin colour [50, 66, 88, 105, 108, 140] or head [93] and human [54] form.

Initialisation is not just detection, but also establishing all of the parameters of the model of the target as it moves through the scene. This initialisation is especially hard for 3D models, since 3D articulated human models have fixed parameters (*e.g.* limb lengths) and 12–43 parameters to track (table 2.5). Although tracking is an activity over multiple frames, initialisation is usually based on a single frame [126, 137].

The extension from a single to multiple targets is very difficult, since the *number of targets* may not be known and varies with time. There are missed detections where the target is not observed, and observations may be false alarms due to clutter, which is of a quite different nature from that inherent in, for example, radar tracking.

Most systems are resource limited. Therefore, when multiple object tracking is performed, the object models tend to be simple. They are typically 2D blobs [54, 65, 72, 74, 86, 88, 105, 107, 133, 140, 142], 3D blobs with an added depth parameter [17, 146], a simple articulated model [134, 153, 164] or implicit shapes [52, 93, 94].

The interaction between objects and the scene is ignored [65,74,107,133], although with an extended mean shift [86] it has been possible to handle simple or multiple occlusions by maintaining an identity map. When multiple objects occupy the same region of the map, the maximum value of the similarity function can judge which object is visible and which is occluded.

In very crowded scenes, separate human poses cannot usually be recovered, since tracking large numbers of objects, with continuous occlusion is hard, therefore global features such as crowd optical flow encoded by HMM [165] or the high energy content of tracked feature points [80] define behavioural patterns. In such crowded scenes, a potential solution is to apply a gross detection of motion pattern, then use a PTZ camera zoom on this. The second level performs pose recovery and detailed behaviour analysis, which is

an integration problem beyond the current state of the art.

### 2.3.2 Human dynamics

Human dynamics define the configuration of the human model based on the prior configurations. There are two levels of human dynamics: the first represents the global translation of the whole body and the second the pose dynamics of the body parts. The majority of work considers the most generic motion model for global position, that is a zero or first order model, with a *zero mean Gaussian* distribution of parameter velocity or acceleration. Tuning the modelling distributions allows any motion, however this might lose important prior knowledge in a constrained environment. W4 [52] uses a first order motion model[2] that reduces the search space by predicting the probable locations. In each frame the new initial position is estimated and then adjusted by a silhouette matching algorithm, maximising the correlation in a small neighbourhood around the initial position.

In a more elaborate model, Antonini *et al*. [140] use a *discrete choice model* consisting of a choice set, a set of attributes, a set of social-economic characterisation (*i.e*. a utility function) and a random term. The model of an individual defines the choice from a set of 33 discrete alternatives of speed and position changes. The parameters of the model are learnt from real data.

For pose dynamics, joint angle dynamics are modelled by a *zero* [121,133,136] or *first* [26,27,108,116] order *Gaussian model* (0GM, 1GM). Green and Guan [26] assume constant angular velocity for the joint angles with each being constrained by anatomical limits, body part inter-penetration avoidance and joint angle equilibrium position. Sidenbladh *et al*. [121] use a zero-order Gaussian model as well as the learnt model with multivariate principal component analysis, where the prediction is a function of the Eigen-coefficients and the walking phase, both with Gaussian variation. In a more constraining model for periodic motion, the priors of Ning *et al*. [126] are learnt from the periodic joint angles scaled into the period of the variation, while each sample of the period has a Gaussian distribution.

Certain *HMM models* learn the transitions between discretized poses. Lan *et al*. [155] represent the walking human poses with a 26-state HMM with groups of one to four states for specific views, achieving robust human tracking in walking. However conversion to other actions needs redesign of the HMM. The main problems of this approach are low

[2]in their definition this is second order model

processing speed and tracking in two dimensional space only. Brand and Hertzman [166] use *Style machines*, *Stylistic* HMM (SHMM), to automatically learn motion. SHMM states are generated minimising the entropy of the model. The HMMs are generalised, characterised by a parameter vector, which restricts the SHMM to a unique HMM. The SHMM is learnt from 50–70 seconds of motion capture data, minimising the compounded entropies of the data using EM. The SHMM was able to represent locomotion well, proved by the ability of the system in identifying and generating motions.

Since human motion depends on longer term history, it is arguable that HMM modelling is not sufficient. Sminchisescu *et al.* [167] use a Maximum Entropy Markov Model (MEMM) and Conditional Random Fields (CRF) that can generate the observation using any length of hidden states. Both methods can be implemented efficiently with dynamic programming. Using 2D and 3D features Sminchisescu *et al.* show that HMMs are outperformed by MEMMs that usually give poorer results than CRFs. Using the same line of reasoning as HMMs, Taycher *et al.* [114] use *transition probabilities* between similar cells (*i.e. key poses*) as a prior for their grid tracker. Assa *et al.* [168] tackle the extraction of key poses relevant to a human observer for video segmentation and computer animation. They extract key frames after transforming with Replicated Multi Dimensional Scaling (RMDS) the high dimensional motion into a low, 5–8 dimensional motion curve. The local extremal points with maximised distances between them correspond to key poses. Appropriate design of the affinity matrices used by RMDS allows extraction of poses both from 3D models and static or moving camera videos.

Since actions differ, several authors have proposed multiple models for different activities, using a *switching model* when one is more accurate than the other. Pavlović *et al.* [169] use a learnt switching linear dynamic system model for modelling human motion. The motion model is linear in the parameter ranges, and the transition between these is a learnt Markov process with transitional probabilities $\Pi$:

$$\mathrm{x}_{t+1} = \mathrm{A}(s_{t+1})\mathrm{x}_t + \mathrm{v}_{t+1}(\mathrm{s}_{t+1}), \tag{2.3}$$

$$\mathrm{y}_t = \mathrm{C}\mathrm{x}_t + \mathrm{w}_t, \tag{2.4}$$

$$\mathcal{P}(s_{t+1} = i | s_t = j) = \Pi(i, j), \tag{2.5}$$

where $\mathrm{x}_t$ is the hidden state, $\mathrm{v}_t$ is the process noise, $\mathrm{y}_t$ the measurement and $\mathrm{w}_t$ the

measurement noise. This model is trained with a one, two or four-state HMM, each with a zero or first order linear dynamic model for walking and running of an 8 DOF 2D human. The trained model generates and segments walking and running motions. Pavlović shows the power of a switching model in a high dimensional and non-linear system, though the extensibility to real, more than 8 DOF and non repetitive action model is not known. Zhao *et al.* [129] also use a switching model, with piecewise linear components. Transitional probabilities define the sequence of model switches. Zhang *et al.* [132] represent the contour of the human with five switching basic models, each from different fixed viewpoints.

Techniques borrowed from *texture analysis and synthesis* aim for better motion modelling. Periodic motion data (*i.e.* joint angles) are synthesised by decomposing the training data into frequency bands, estimating their distribution with kernel-based techniques, and sampling them for synthetic motion generation. Pullen and Bregler [170] use this method for low dimensional data generation, while extension to higher dimensional data needs further examination. Liu *et al.* [171, 172] learn dynamic textures by EM and include them in a mixed, probabilistic PCA model, then globally align the high dimensional data into a single manifold. Trained on human articulated motion [172], synthetic motion is generated, though it could have trouble with non-linear motion and switching activities. Sidenbladh *et al.* [122] build a 3D motion model by structuring the whole training data from motion capture into a binary tree-like structure, using 16D PCA, the testing condition being the PCA coefficient sign. The last frames define the possible next configuration using texture synthesis. The temperature parameter determines the variance of generated model, and can be used to allow larger changes from one frame to the other. Accessing the whole training data makes the estimate robust to noise; the tree structure favours quick computation. The model can be extended simply by adding new sequences, though generalisation and scalability of the model is limited, since the similarities with the already known training data are ignored.

*Physics-based models*, increase the accuracy, stability, and generality for person tracking, as Brubaker *et al.* have shown [173] with their simple walking lower limb dynamic model. However it is doubtful that all the interactions involved in an complex articulated model within an arbitrary environment can be effectively modelled.

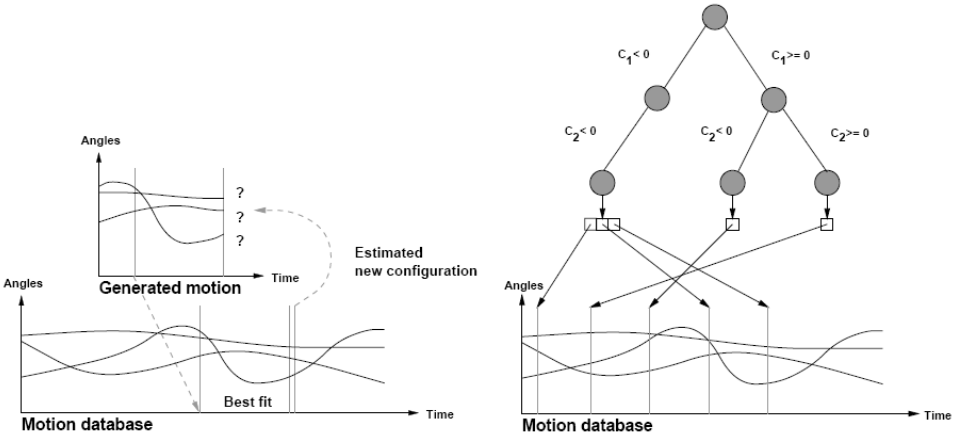



Figure 2.7: Motion prediction and generation (left) with database lookup with tree search (right) from [122].

### 2.3.3 Scene models

#### Camera models

The exact properties of image formation performed by CCTV or other cameras capturing scenes is complex and the physical characteristics of the camera architecture are not always known. Therefore orthographic and full or simplified Tsai [174] perspective models are used to approximate the real process. Since tracking does not require accurate measurement simpler models are usually quite adequate. The scaled orthographic or weak perspective camera model is preferred [100, 118, 119, 153] since it has fewer computational needs, requiring only a single scaling by the object depth. However, it fails to model the depth related perspective changes in many real scenes, such as that illustrated in the i-LIDS underground station data set [175]. In those cases a full perspective model is preferred [120].

#### The ground plane

Requiring camera calibration, the *Ground Plane Constraint* (GPC) [134,140,146] is usually a very strong prior, and during tracking target location is maintained as the 2D coordinates of the flat ground plane, since humans translate maintaining permanent contact with the ground, and with constant height. From these assumptions, with a few tracked parameters, the whole human model can be parametrised. As another example, Remagnino *et al.* [93] used the ground plane as a prior calibration to track 3D car and humans models projected onto the image plane.


| *Symbol* | *Object* |
|---|---|
| CA | Central area |
| FA | Front area |
| M1 | Left hand side machine |
| M2 | Centre machine |
| M3 | Right hand side machine |
| DBC | Back centre door |
| DBR | Back right hand side door |
| DMR | Middle right hand side door |
| DBL | Back left hand side door |
| NBR | Right hand side notice board |
| NBL | Left hand side notice board |
| C1 | Chair |
| C2 | Chair |
| C3 | Chair |
| C4 | Chair |
| WC | Wall clock |
| PC | Photocopier |

Figure 2.8: A scene model example

### Scene configuration

In addition to the flat ground plane assumption, more complex scene models further improve the tracking. For example, to date, as far as we are aware, stairs or other oblique surfaces have not been modelled as a prior in a human tracking applications, although 3D environmental models [134] have been used to aid reasoning about occlusion. The scene defines areas where humans can go easily and where they are found more frequently. Johnson and Hogg [176] learn by the mean of a neural network the probabilistic distribution of pedestrian flows, tracks generated by [92]. This allows prediction of the future segments of the pedestrian paths in a scene and the detection of unseen (*i.e.* suspicious) path patterns.

Entrance and exit areas or otherwise defined regions where the object is instantiated or eliminated [17, 105, 140, 142], reduces the exhaustive search for new objects.

Complex scene models, such as the one in figure 2.8, define not only the allowed spaces also defines the behavioural context, *i.e.* the activities that can occur only in specific regions of the scene.

### Background modelling

Wang *et al.* [150] perform background subtraction from median background pixels computed over the last $N = 60$ frames. The R, G and B channels are considered independently and a pixel is considered as background if so defined in any of the channels. Krumm *et al.* [17] model the background with a single mode of a Gaussian of a 3D depth image for an

indoor scene. Similarly, the Gaussian background model of Wren *et al.* [106] is recursively updated with an adaptive filter.

For outdoor scenes with more rapid background fluctuation, Haritaoglu *et al.* [52] use a bimodal background model with the minimum, maximum pixel values, and the maximum difference between consecutive intensities. The model is updated both on a pixel level to equalise global changes, and on an object level to incorporate into the background new or removed objects.

Stauffer and Grimson [177] define the background model as a mixture of $K$ Gaussian distributions of the history of each pixel $\{\mathrm{X}_1, \ldots, \mathrm{X}_t\}$ preceding the current time $t$:

$$\mathcal{P}(\mathrm{X}_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(\mathrm{X}_t, \mu_{i,t}, \Sigma_{i,t}), \tag{2.6}$$

where $\omega_{i,t}$ is the weight of the $i^{th}$ Gaussian with mean $\mu_{i,t}$ and variance $\Sigma_{i,t}$, and density $\eta$:

$$\eta(\mathrm{X}, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathrm{X}-\mu)^T \Sigma^{-1} (\mathrm{X}-\mu)}. \tag{2.7}$$

To detect foreground pixels within a new image, each pixel is first checked against each of the $K$ distributions until a match is found. If none of the distributions match the pixels, then the least probable distribution is replaced with the current pixel. If a match is found, then the weights $\omega_{i,t}$ are adjusted. Distributions are ordered in decreasing order of their evidence/variance ratio and if the pixel is part of the first $B$ distribution defined by their summed weight being over a threshold $T$, then the pixel is background, otherwise it is foreground. The method is fast, with 11 to 13fps (on an SGI O2 with a R10000 processor) for a $160 \times 120$ pixel resolution image, and robust, adaptive to weather conditions, and has been evaluated continuously in trials over 16 months. The Gaussian mixture background model is also used effectively by Sigal *et al.* [137], Ning *et al.* [126], and Porikli and Tuzel [65], with the later performing an update only for the high variance pixels in the mixture, when large changes are detected, *e.g.* caused by illumination variations. Despite its wide acceptance, Hall *et al.* [178] compared five different background subtraction methods and suggest that the complexity of [177] is not fully motivated compared to other simple methods with better tracking detection rate and false positive rate.

Colombo *et al.* [179] present a HMM based background model, able to model periodic periodic changes in indoor scenes (*e.g.* moving staircase), with higher accuracy compared

to the Gaussian mixture model of Stauffer and Grimson, since the involved $N$ states model the periodicity of the transitions.

Liu *et al.* [180] use a mean shift over pixel values over several time steps, which performs well for sequences with non-stationary objects. The method does not require parameter tuning, which is a substantial advantage.

Spagnolo *et al.* [181] uses a background model based on motion detection computed from differences between two consecutive frames. For motion detection, and for image to background model comparison, the radiometric similarity is evaluated in small windows around the current pixel. As the comparison is window, not pixel based, it aims to be robust to noise. Sudden environmental and illumination changes are detected if the percentage of the total foreground pixels exceeds a threshold (40%).

Kumar *et al.* [182] also use motion to segment moving foreground objects. It finds the moving layers, parts that have coherent motion, of all foreground objects by a multi step optimisation, recovering first the approximate transformation parameters, optimising them with loopy belief propagation. Analysing rigid parts over the whole sequence, rigid segments of the deforming body are recovered on a coarse scale, then enhanced using colour information and graph cut energy minimisation. This method has potential for moving cameras, as the background can be expected to result in a single motion layer due to the affine camera transformation. However, the deformable layered body part reconstruction may fail when the motion is not in the fronto-parallel plane, with large viewing angle changes.

#### Shadows, reflections and occlusion

*Shadows* and *reflections* [88, 99, 106, 156] cast onto the ground or other objects are often recognised as ghostly moving objects. The usual approach to shadow removal is to analyse the colour space model. For example, they are eliminated in HUV space by decreasing the luminance and changing the saturation of the background pixel [88]. Similarly, Wren *et al.* [106] use the brightness normalised chrominance to detect the shadow pixels. Zhao *et al.* [156] in the 3D space hypothesise not only the human, but also the casted shadow and therefore consider their reciprocal influence.

*Occlusion* is a major factor in tracking loss, especially with multiple targets, since the problem of data association is manifest. Even when tracking an isolated object, *e.g.* when using an articulated model, self occlusion occurs when body parts occlude other

body parts, causing ambiguous configurations. Multiple camera tracking [117, 137, 141] is the general solution to this problem. Environmental priors, for example defining common progression routes in the scene can to some extent resolve the ambiguities.

### 2.3.4 Behavioural priors

General knowledge of how people behave in usual and strange situation helps to detect suspicious behaviour. Possible patterns are provided by the psychological literature.

Behavioural observations that have a high (30%–55%) probability of occurring arise from appearance, body language, and proximity to high risk goods. Factors with low (under 30%) incidence rates include age, known offences, inclusion within a gang or group, looking at the cameras, ethnicity, and single motherhood [183]. Further, shoplifting behaviour studies [4,5] have tried to reveal features that can help focus attention on suspicious individuals as subjects of tracking. Buckle and Farrington [5] claim that more women than men are shoplifting, that elderly and young groups (*i.e.* above 55 and below 17) are more likely to offend. This is in strong contradiction to Dabney *et al.* [4], who puts higher odds of shoplifting within the 35 to 54 years age group. Dabney *et al.* also conclude that features such as race, age, sex, and season of the year are not important factors in shoplifting behaviour. He noticed that shoplifting for those exiting without buying a product was six times higher that those who bought something, while Buckle and Farrington [5] observed that most shoplifters purchased items. These observations are very contradictory, and may be due to the different profile of the observed stores (department store [5] and pharmacy [4]), or indeed many other complicating factors. Given the uncertainty on the utility of the above features and since these features are extremely difficult to detect by means of computer vision, at the current state of the art, they can be ignored. However, there is some common ground between these two studies. Buckle and Farrington, as well as Dabney *et al.* concur on the following shoplifting clues:

- scan, tamper, display awareness of security countermeasures, sample products [4],
- increased attention: checking if nobody is watching them [5],
- change in attitude: talkative changes to silent behaviour,
- usually they pick up the object and while checking other objects they hide the object to be stolen.

Though suggestive, the pre-shoplifting behaviour was reported in 19.9% of the total shoplifting cases [4], therefore they can raise awareness of the supervising staff, but are not enough for decision making, hence the actual act must be detected.

These pre-shoplifting observations are not yet successfully incorporated in any of the automated analysis systems for the reasons that their definition is not yet well formalised, and that they require a fine detail of analysis and reasoning that current algorithms cannot detect.

### 2.3.5 Conclusions

The models for humans are diverse and include 1D shape contours, 2D blob representations, 2D shapes, 2D articulated models, 3D blobs and 3D articulated models. These models are strongly tied to particular applications.

In general, the majority of work on tracking and behavioural analysis has been based on static video cameras, in part because of their wide deployment and low cost, in part because that problem is more tractable when compared with the analysis of video footage from a moving camera. In particular, this allows background subtraction to extract moving objects, while 3D scene configuration is recommended for 3D model tracking.

2D human models are simpler than 3D models, and are regarded as the first choice when single view input is available. However, view dependency, scaling and occlusion handing are hard to handle, and these aspects are implicitly solved by 3D models.

Blob models have only a few parameters, therefore they can be recovered with high accuracy and present a low challenge. However, articulated models fit the image data better and provide detailed input for behaviour analysis at the cost of larger parameter space.

Low level information derived from blobs, such as position and speed, allow simple interactions to be detected, such as the detection of an abandoned object, periodic activity or gait recognition. 3D blobs have additional depth information, therefore positional parameters provide more exact localisation.

Where the classifiable behaviour is also defined by the limbs (*e.g.* reaching, picking up, carrying, handshaking, *etc.*) an articulated model is usually required. However, there is an alternative in the statistical or other analysis of the space-time surface as the human moves through space.

Dynamic models generally use zero or first-order Gaussian motion models. These

model arbitrary motion satisfactorily. However humans have a set of (periodic) motion patterns (*i.e.* walking, reaching), and therefore a model with switching between these patterns is plausible. While modelling individual activities independently improves the prediction, it also makes the model inflexible and hard to extend. Model completion was shown to be powerful, although it has questionable extensibility. Therefore, for flexibility and adaptability it is necessary to discretize motion into sets and to implement a switching between these sets.

## 2.4 Image measurements and likelihood functions

All vision based algorithms process the input raw images, using standard or modified methods for extracting edge, foreground silhouette, colour, texture or motion. These reduce the high dimensional input image to a set of edges, blobs or feature vectors. Ultimately, these characterise how well the model fits the input.

Within a Bayesian approach, equation (2.2), $\lambda(\mathrm{O}|\mathrm{x})$ is the likelihood of observation O with respect to the parameter x. It expresses the image based measurements and according to Duda [184, p.22] is *indicating that if other things are equal, the category* x *for which* $\lambda(\mathrm{O}|\mathrm{x})$ *is large is more likely to be the true category.*

Approaches to evaluation of the likelihood in image data are many, and table 2.7 gives an overview of variants and their relation to the applicable priors and methodology. Further, table 2.9 focuses on different sources of measurements and their combination for Bayesian techniques, mainly using PFs. The most frequent is the edge likelihood, *i.e.* the distance of the projected model edges to the image edges, or the less stable binary match. Similarly, the silhouette likelihood compares the projected model with the silhouette image, usually generated by background extraction. Colour likelihoods either use colour histograms with multiple or a single (mean) bin, or form a match with a predefined (*i.e.* skin) model. Appearance (texture) [115, 129] and thicker edges (ridges) [185] are other alternatives. The majority of authors consider the body as a whole, and compute a global likelihood, which is less sensitive to limb deformations. Likelihoods based on parts, and assembled into global likelihoods or used directly [128], provide a close fit and faster convergence of the algorithms. All of the part based likelihoods consider the individual likelihoods to be independent. Although this must be a considerable flaw, the computations are otherwise intractable. If multiple camera views are available, they are

also considered independent.

| Author | Likelihood | | | | Body | Camera |
|---|---|---|---|---|---|---|
| | Edge | Silhouette | Colour | Other | parts | views |
| Bălan *et al.* [141] | distance | match | – | – | whole | multi |
| Deutscher *et al.* [117, 139] | distance | match | – | – | whole | multi |
| Green and Guan [26] | distance | match | – | – | whole | single |
| Kang *et al.* [186] | histogram | – | histogram | – | whole | single |
| Kim *et al.* [125] | histogram | – | – | – | whole | single |
| MacCormick and Isard [128] | distance | – | – | – | parts | single |
| Ning *et al.* [126] | match | – | – | – | whole | single |
| Lanz [112] | – | – | histogram | – | parts | single |
| Okuma *et al.* [110] | – | – | histogram | – | whole | single |
| Osawa *et al.* [134] | – | match | – | – | whole | multi |
| Pantrigo *et al.* [127] | match | – | – | – | whole | single |
| Saboune and Charpillet [111] | – | match | – | – | whole | multi |
| Shan [108] | – | – | mean | motion | whole | single |
| Sidenbladh *et al.* [120, 185] | learnt | – | – | ridge, motion | whole | single |
| Sigal *et al.* [137] | – | – | – | learnt texture | parts | multi |
| Taycher *et al.* [114] | – | – | – | texture (edge histogram) | whole | single |
| Sminchisescu and Triggs [124] | distance | match | – | – | whole | single |
| Sminchisescu and Triggs [115] | distance | – | – | texture | whole | single |
| Stenger *et al.* [113, 148] | distance | – | skin colour | – | whole | single |
| Sudderth *et al.* [136] | distance | – | skin colour | – | parts | single |
| Wu and Yu [149] | histogram | – | – | – | whole | single |
| Zhang *et al.* [132] | distance | – | histogram | skin colour | parts | single |
| Zhao *et al.* [129] | distance | – | – | texture (colour histogram) | whole | single |
| Zhao *et al.* [107] | – | – | histogram | | parts | single |

Table 2.9: Likelihood for generative approaches.

Well designed likelihoods are important, since as Sminchisescu [124] also remarks,

problems frequently arise from unbounded spurious peaks, and singular, flat or very low curved likelihoods. The major forms of image data used to compute the likelihoods are reviewed in this section, with the observation that variants can be combined without limit.

### 2.4.1 Silhouette

Background model subtraction (section 2.3.3) is simple to perform and results in foreground regions, or silhouettes of objects, that are a most important clue for tracking. This is first because most of the interest lies in foreground object tracking, and second because the foreground area is usually smaller than the background, reducing the tracking search space. Given the extraction of silhouettes, a likelihood function based solely on the binary data is straightforward [117]. Pixels of the estimated model are matched against the foreground silhouettes and the matching score results in the likelihood. As an alternative likelihood, Saboune and Charpillet [111] use the ratio of the common pixels and the sum of model and silhouette only pixels. Sminchisescu [124] compute a silhouette likelihood that consists of a term maximising the overlap of prediction and observed silhouette and a term pushing the model inside the image silhouette. For effective computation, the Chamfer distance transform is used. Lv and Nevatia [41], and Howe [98] compute likelihoods from shape context descriptors that encode silhouettes.

Silhouettes are useful for presence detection and object localisation, but they ignore the internal features of the object, and for a deformable object are poor in characterising the identity. Therefore additional measurements are required.

### 2.4.2 Edge

Edges define boundaries between regions and may be external or internal. External edges are between object and background, or between multiple objects, and internal edges are between parts of the tracked object. They are robust against illumination variation and improve localisation [116], but are computationally expensive [133] and are less stable than region based likelihoods. Kim [125], MacCormick and Isard [128], Stenger, Thayananthan *et al.* [113, 148], and Gavrila [63, 76] use edges only, while other authors combine them with additional features [124].

For fast computation, a Sobel detector can be used [124], but Canny edges provide better accuracy [113, 129, 148]. Similar to silhouettes, either direct counting of the matched

silhouette point likelihoods [117,126] or the distance from the edge [128] provides the likelihood value. For the latter, the fast Chamfer distance transform is used [63,76,113,148]. For example, Gavrila [63,76] uses the Chamfer distance transform to compute similarities between edge image features for template matching. Zhao *et al.* [129] model only at the two ends of the segment with forces increasing with distance from the edge. Sidenbladh and Black [185] use edge (first order derivative of the input) and ridges (second order derivative) likelihoods, trained on body and non-body images. Similarly, Sigal *et al.* [137] apply multi-scale edge and ridge likelihoods, also modelling conditional dependencies between different filter responses.

Edge histograms, as used by Kang *et al.* [186], with learnt edge distribution compared to edge matching, also capture the texture or structure of the tracked objects.

Edge information is preferable to silhouettes when foreground extraction is difficult, *i.e.* when the camera is moving or environment changes significantly.

### 2.4.3 Colour

Colour likelihoods express the probability of pixel or region colour being in an a-priori specified colour range, or the probability of a match with an a-priori appearance model learnt from a training database or initialised in the first frame, whether updated or not during the tracking.

The obvious colour range in human tracking is the colour of the skin that characterises the face [105], hands [187] or both [50, 88, 148]. The likelihood is given by the learnt probability of a colour being skin [88,105,187] or the probability distribution of skin and background [50,148,187] learnt in RGB [88], normalised chrominance [105] or other spaces, modelled with a Gaussian or Gaussian-mixture model. The choice of the colour space and model varies from author to author.

Region model based likelihoods use a distance between the region colour histogram and the histogram of the model. They differ in the colour distance computation and in the colour space used. The Bhattacharyya distance is the most frequently used dissimilarity metric [110,112], but the Kolmogorov-Smirnov distance [188], Kullback-Leiber divergence [129], Mahanabolis distance [106], quadratic colour histogram measure [189], or $L_1$ metric [190] are also applied. The major difference in such metrics is whether they allow cross-bin comparisons, necessary to account for shifts in colour space caused for example by changing illumination, or whether they only allow bin-to-bin comparisons. For histogram

formation a method apart is the perceptually weighted histogram [191], with each pixel contributing not only to a single colour, but to the most similar ten bins. Several colour spaces have been used, including RGB [65], normalised YUV [106], HSV [110], CIEL*u*v [191], CIELUV [189], S-CIELAB [188, 190]. The invariant spaces are favoured, as an example the S-CIELAB colour space [190] has perceptual meaning, formed by Gaussian smoothing with different kernel sizes of the intermediate CIEL-XYV space. There is no general acceptance of which colour space is the *best*, as it depends on the actual application. Articulated objects, with different colour distributions of the parts, might be described with multiple models for head, upper and lower body [65].

Colour histograms are robust against noise and partial occlusion, but as suggested above by the necessity for cross-bin comparisons, they suffer from illumination changes or confusing or changing background colour, ignoring partial layout configuration [133]. They are also computationally expensive when there are large regions and numbers of samples.

### 2.4.4 Texture

The textures of 3D objects are unchanged during tracking and provide strong spatial 3D measurements. They also incorporate colour, silhouette and edge features, although their representation and the computations involved are more complex than for other features. Sidenbladh *et al.* [121] use unwrapped limb textures compressed by weighted principal component analysis, that with EM computes an orthogonal transformation of the training data that might have occluded, incomplete textures. Examples of other texture likelihoods include the four level 2D wavelet transform of Kam *et al.* [188] evaluated with the Kolmogorov-Smirnov distance or the multiple colour histograms of Zhao *et al.* [129].

### 2.4.5 Multiple camera observation

Visual information can also be provided by multiple sources. The VSAM uses multiple cameras to track humans [164]. To register two non-overlapping views by affine and perspective transform, a PTZ camera is used by Kang *et al.* [186], and then a joint spatio-temporal model is used to track people across multiple views to tackle handover. Tracking over multiple frames can be solved by integrating across multiple cameras, example for face and number plate recognition in IBM's S3 system [75].

Multiple camera tracking provides more detailed information about 3D articulated structure [26, 77, 117, 119, 137] or can be more reliable for extracting 3D position [17, 134, 146], since self occlusions are reduced, and multiple measurements provide more rigorous data.

### 2.4.6 Non-visual observation

So far, the likelihoods were limited to regular visual input, however non-visual sensors, or post processing produces other observation modes of the scene. Hence, other likelihood functions are based on the disparity map of the hand and the head [50], ultrasonic 3D tags localisation [192] or motion [65, 185] resulting from inter-frame differencing.

This thesis considers only scenes that are observed by a single type of sensor, notably a standard video or CCTV camera or cameras. There is an obvious advantage in this because of the existing widespread deployment of such cameras; a robust algorithm can be widely applied solely with the addition of computational power. However, there may well be cases where multi-modal data can improve the likelihood of accurate scene analysis. Further, many of the algorithms deployed for video data are equally applicable to the IR case, for example. In terms of human behavioural analysis in cluttered backgrounds, there have been some attempts to combine video and audio data. CASSANDRA [80] is one example of a multi-modal system, fusing audio and video clues with a Dynamic Bayesian Network. Pitch and spectral tilt define the energy of an audio channel that warns of aggression by ergotropic arousal. The energy of tracked KLT feature points is analysed for video based aggression. In a process analogous to background subtraction, the audio channels have to be filtered to exclude environmental noise such as that caused by passing trains. Böhme *et al*. [66] combine visual colour and head-shoulder contour clues with sonar and audio measurements for better localisation with changing lighting and scene background. Brdiczka *et al*. [49] also combine visual clues with speech detector to detect human interactions. In KidsRoom [142] three cameras perform specific tasks: an aerial-view camera performs human tracking, and two room level cameras recognise body movements at two specific locations. A microphone, by voice recognition, facilitates interaction with the system.

### 2.4.7 Measurement fusion

Despite the frequency of edge based likelihoods (table 2.9), Bălan *et al.* [141] show that likelihood form silhouettes are more powerful then from edges, though they fail to identify limbs occluding other body parts (*i.e.* arm in front of the torso). Colour likelihoods are stable in tracking [84], but need initialisation

In future, to achieve robustness, the combination of multiple measurements is required, since all likelihoods have assets and handicaps:

- silhouettes are easy to compute, since they are region based they are less sensitive to noise, they can be used without initialisation, but are poor in clutter and cannot maintain identity or internal configuration,
- edges discriminate borders between multiple foreground objects and parts of the foreground objects, and use a simple a-priori model without or with a simple initialisation, but edges are more complex to compute and more sensitive to noise,
- colour, an extension of the silhouette model, allows higher discrimination and is able to maintain the identity of the object, however it is more sensitive to noise and environmental changes, and requires initialisation,
- texture has many of the advantages of silhouettes, edges and colour, but can still be noise-sensitive, and have a cost of complex representation and computation, requiring initialisation.

Although possibly the most discriminatory, textures are not frequently used in tracking, perhaps because most objects tracked at low resolution have no obviously defined texture. Against this, the other components can be combined in several ways. Since occlusion between limbs generates problems, Ning *et al.* [126] combine edges and silhouettes. Zhang *et al.* [132] use independent edge gradient, silhouette, skin colour and region similarity clues. Sidenbladh and Black [185] combine edge, ridges and motion (predicted image histogram based on the previous image and the motion model) computed from foreground and background likelihood ratios. Sidenbladh and Black conclude that motion likelihood has the most discriminatory power, followed by edge and shape. Wachter and Nagel [116] use both edges and regions for model fitting, because in general region information stabilises the tracking, while edges improve localisation.

Lanz [112], performs a fusion of likelihoods of parts. He computes the colour likelihoods for body parts, and then the global log-likelihood of a pose, as the mean of the body part based likelihoods.

Choosing one or other of the likelihood functions affects tracking performance seriously and yet it is still an art of the algorithm designer rather than a science.

## 2.5 Human tracking systems

This section reviews large tracking systems that contrast with other tracking algorithms in being complete as standalone applications, and in providing an abstract representation of the processed video stream.

Pfinder [106] is one of the earliest real time tracking and behaviour analysis systems. Using spatial and textural features of the pixels, a 2D blob model is initialised with head, hands and feet and other clothing, when a person is first detected. These are clustered into blobs with similar properties, later tracked using a maximum a posteriori probability approach. Through a modular interface, features of the tracked blobs are provided for further interpretation such as sign language recognition or game and virtual systems interactions. Pfinder is robust to environmental changes and occlusions, because of the automatic re-initialisation process when current tracking it lost. However, it has limitations: it is applicable only to a single person in view, and to environments with less dynamics in the background than in the foreground. Gesture recognition works well for the designated applications, however it is not generic and has a restricted vocabulary.

The Easyliving environment [17] was designed for domestic applications, performing robust blob tracking of 1–3 persons. Easyliving has two systems of three cameras to generate the registered depth and colour images for person segmentation and 3D localisation. Blobs are tracked frame to frame, and their identities are maintained within a ground plane map with each cell having a colour cube histogram. Occlusion problems are solved by the two sets of stereo cameras. The predefined entry and exit areas simplify the detection phase. Easy-living demonstrates that a simple detection based approach can accomplish behaviour understanding for a homelike environment.

The mobile robot of Böhme *et al.* [66] is an interactive shopping assistant and a mobile information kiosk for customers. The complex hardware used for localisation includes a omni-directional colour camera with panoramic view, two pan-tilt-zoom colour cameras,

a microphone pair, two layers of 24 sensor sonar, two PCs and a touch-screen. This seems rather expensive in comparison with the functions offered by the robot.

The Bobick *et al.* [142] system is an intelligent playground, tested in real life, implemented with programmed rules. Separate algorithms are designed for blob tracking. Using overhead cameras, they use background subtraction, colour, velocity and size information to disambiguate objects; and action detection provided by the blob dynamics shape (*e.g.* crunching), pose (*e.g.* arm in Y shape) and movement (*e.g.* spinning).

Video Surveillance and Monitoring (VSAM) [164, 193] was a three year project (1997–1999) conducted at the Robotic Institute of CMU in cooperation with the Sarnoff Corporation. The purpose was to minimise human intervention in surveillance, that is to automate behavioural monitoring to as great an extent as possible, but it stopped short of full automatic analysis. A single, simple camera system was able to detect people and vehicles, and to classify them as humans (walking or running), human groups, cars or trucks using shape and colour analysis. Classification used a three layered neural network trained for each camera on blob dispersion, the aspect ratio of the bounding box, the area of the blob and the zoom factor. Template matching of intensity models tracks in single view, while histograms match across multiple views. Activity was analysed by the geometry of blobs: walking and running are identified from the gait-periodicity. Markov models are trained for simple action detection such as meeting, vehicle driving and dropping someone. VSAM also deals with multi-sensor problems, such as matching targets by trajectory and normalised colour histogram; and camera alignment for view optimisation. The authors report good real time performances in the given surveillance scenario. The main disadvantage of VSAM is its complexity. It requires high specification hardware, and is not flexible for other surveillance scenarios requiring detailed analysis of humans and their interaction.

IBM's Smart Surveillance System (S3) [75] uses Adaboost for face recognition, number plate recognition, object colour and size classification, object location, extraction of dynamic parameters and event duration. Then events can be defined by the combination of these features in the database.

Research at the Universities of Leeds and Reading provided a series of results in tracking for surveillance. Baumberg and Hogg's tracker [92] uses an active shape model with reduced dimensionality (14 DOF) for tracking the walking human contour with B-Splines. Real time performance at 30fps is achieved on a 100MHz R4000 SG Indigo workstation,

however the outlines with hands apart from the body are not tracked well. A possible higher dimensional model would solve this problem, but with increased processing time. Combining the Leeds University People Tracker and the Reading University 3D rigid object tracker for cars resulted in an Integrated Traffic and Pedestrian Vision System [93]. The system can analyse behavioural interactions by matching spatio-temporal trajectories to a probabilistic model; with Bayesian belief nets a natural language description of object interactions was generated. The two subsystems track semi-independently, exchanging information about the occluded regions. As part of the Advisor project, Siebel's system [94] fuses Baumberg's active shape tracker with a blob based tracker into a flexible and expandable application with high robustness. The two trackers, figure 2.9, can maintain correct tracking in more complex situations such as the underground station surveillance scenario test bed of the Advisor project.

Figure 2.9: Reading tracker [94] run on a CAVIAR sequence. On the first line four frames from *OneStopNoEnter2* sequence show the successful detection over multiple frame, however the altering colours of the contour are incorrect identity changes. The second line shows failed detection from *EnterExitCrossingPaths1* and *TwoEnterShop1* sequences with false positives and negatives, with multiple humans detected as a single human and with incorrect contour detection.

The W4 system [52] is one of the most complex surveillance systems. It simultaneously tracks a number of independent and grouped people, detects posture and actions such as carrying objects and leaving them behind. The system has several subsystems performing: foreground segmentation, people detection, individual tracking, body part detection, group tracking, and unusual activity detection. It has no unique framework, each subsystem is empirically adjusted and applies computationally simple methods. They report reliable and on-line results.

## 2.6 Training and evaluation data

As yet, there are no universally agreed benchmark datasets for the evaluation of the various methods, although the HumanEva and i-LIDS datasets described below are attempts to do just this. Rather, each developer defines his/her own requirements of the video data. The main differences consist of the quality and the length of the visual data, and whether ground truth for training and evaluation is available. Although video recording is straightforward, the complementary ground truth acquisition requires either additional costly sensor for *MOtion CAPture* (MOCAP) or laborious manual post processing.

An accurate and objective evaluation requires large datasets, however these are costly and therefore are missing. The data is expected to be real, while scripted scenarios are recognisably different from real events (*e.g.* the BEHAVE scenarios are obviously staged). The several projects below focus their efforts on the acquisition of these datasets, providing a wide, but yet incomplete realm for algorithm development.

### 2.6.1 HumanEva dataset

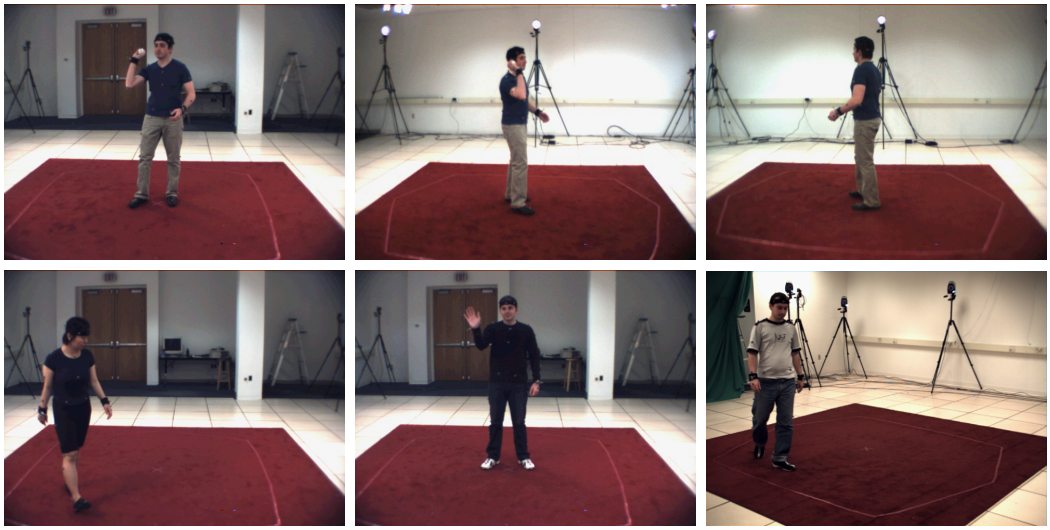


Figure 2.10: Example frames of the HumanEva dataset. The first row shows the three views of a *Throw/Catch* sequence with subject S1, while in the second row subjects S1, S3 and S4 are performing *Walk*, *Gesture* and *Combo* activities respectively.

The HumanEva dataset [194] provides training, validation and test video and MOCAP data, as well as an evaluation methodology for human tracking. The initial HumanEvaI dataset includes *Walk*, *Jog*, *Throw/Catch*, *Gesture* and *Box* sequences, and additional *Combo* sequences that are successive of walking, jogging, and balancing alternately on

each leg (figure 2.10). The sequences are recorded with three colour (C1...C3) four black and white cameras, and with a MOCAP system. By means of MOCAP the 3D position of each body limb is recovered. The sequences are partitioned in three sets, for validation, training and testing. The validation and training sequences have MOCAP data, however the test sequences do not, and the tracking result on these can be evaluated only by the mean of the on-line evaluation system that offers an objective measure of the tracking algorithm. The actions are performed by four subjects (S1...S4) in three trials: the first trial provides the validate and train sequences, the second has test sequences only, while the third trial consists only of training MOCAP sequences without video data. Subject S4 as well as the *Combo* sequence has just trial two, *i.e.* test sequences.

The synchronised video and MOCAP sequences (trial 1 and 2) have 60Hz frame rate, while the MOCAP only trial 3 data, has 120Hz capture rate. Video resolution is VGA quality, *i.e.* $640 \times 480$ pixels.

The HumanEvaII dataset includes three sequences only. The *S1 Walking 1* sequence is frames 6–590 from HumanEvaI, while the *S2 Combo 1* and *S4 Combo 4* contain three sets with increasing complexity: *Walk* (frames 1–350 respectively 2–350), *Walk* and *Jog* (frames 1–700, respectively 2–700), and the full *Walk*, *Jog* and *Balance* sequence (frames 1–1202 respectively 2–1258). Subject S4 has no training data, while for S1 and S2 HumanEvaI can be used. The *Combo* sequences have four (C1...C4) camera views.

### 2.6.2 CAVIAR dataset

Figure 2.11: CAVIAR dataset examples of the INRIA lobby and two views of the shopping hallway.

The CAVIAR data includes two scenarios (figure 2.11). The first is recorded with a wide-angle camera lens in the entrance lobby of the INRIA Labs at Grenoble, France. The second scenario captures a hallway in a shopping centre in Lisbon in two views, one view across and the other along the hallway. CAVIAR has manually marked ground-truth with

bounding rectangles of individuals and groups, and for some sequences body parts are also labelled. The video sequences have half PAL ($384 \times 288$ pixels) resolution at 25fps rate.

For testing our tracking algorithms two CAVIAR sequences were used, each with manually synchronised corridor and frontal views: the *EnterExitCrossingPaths1* sequence contains corridor 0–306 and frontal 76–382, a total of 307 frames, while the *OneLeaveShopReenter1* has 314 frames that are frames 0–313 of the corridor, and 76–389 of the frontal views. The corridor numbering is used when a frame is referred in the thesis.

### 2.6.3 i-LIDS

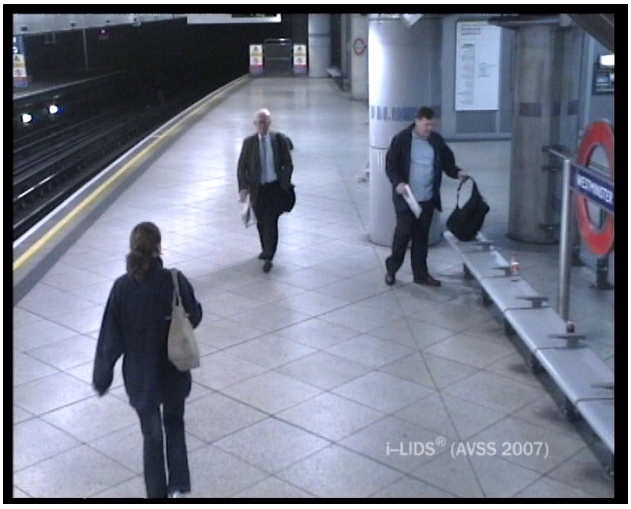



Figure 2.12: i-LIDS Abandoned baggage scenario from the AVSS 2007 medium test sequence

The UK Home Office Scientific Development Branch (HOSDB) designed, recorded and augmented the *Imagery Library for Intelligent Detection Systems* (i-LIDS) video database, to test robustly the end-user (*i.e.* police, security services, other governmental departments) requirements for surveillance algorithms.

At present, the dataset consists of abandoned baggage (figure 2.12), parked vehicle, doorway surveillance and sterile zone scenarios, with clips recorded in PAL-DVD ($720 \times 576$ pixel) resolution at 25fps. The scenarios contain different types of alarm event, with about 250–300 events each, with weather conditions and environmental conditions varying over 12 months recording, *etc.*, and a high-level ground truth for each dataset. The dataset has exact definition of criteria such as the types of objects that are looked for, when they are abandoned, what constitutes abandonment, *etc.* It includes hard environmental

conditions, such as changing lighting from dawn to dusk, rain and snow, night with head lights and low SNR. The dataset is currently being extended for synchronised five camera tracking recorded at London Gatwick airport with varied target, target behaviour, crowd densities and dawn to night lighting conditions.

### 2.6.4 Other tracking datasets

The following datasets are designed to train and evaluate tracking algorithms, mainly in surveillance scenarios, hence focusing on blob-based position recovery and not on behavioural interpretation.

The IEEE workshops on Performance Evaluation of Tracking and Surveillance (PETS) have provided training and evaluation data for tracking for eight years. The most recent dataset (2007) contains three multi-sensor sequences for loitering, theft and unattended luggage. Previous datasets contain multi-sensor sequences of left-luggage scenarios with increasing scene complexity (2006); a subset of the CAVIAR dataset (2004); people interacting with facial expressions, face and hand gestures, and white board activity (ICVS 2003); football players on an outdoor pitch with two views (VS 2003); people moving in front of a shop window (2002); two views of four outdoor scenarios (one with a panoramic camera) with people and vehicles, front and rear views of a moving vehicle (2001); and sequences of people and vehicles in a car-park (2000).

The CLEAR (2007) dataset focuses on 3D Person Tracking, 2D face detection and tracking, person identification, head-pose estimation, acoustic event Detection, 2D Multi Person Tracking, 2D Face Tracking, Vehicle Tracking, the TRECVID (2001–2006) datasets concern about video retrieval, sequences are useful for human tracking and activity recognition.

ETISEO (2005–2007) has scenes from several video surveillance areas with indoor (corridors, building entries, subway stations) and outdoor scenes (streets, parking, airport) on different complexity level and not only visual sensors.

The *Pacific Missile Range Facility* (PMRF) dataset is under development and will contain multi-sensory data, including laser ranger, microwave/infrared, fixed camera, pan-tilt-zoom camera, mega-pixel camera, seismic sensor, radio frequency ID tags sensors recorded on a military base at Kauai, Hawaii.

### 2.6.5 Other behavioural datasets

While track or pose recovery is secondary, datasets from this section allow evaluation of simple behaviour analysis. The BEHAVE dataset comprises of two views of 10 scenarios with 2–5 people performing *in group*, *approach*, *walk together*, *split*, *ignore*, *following*, *chase*, *fight*, *run together* and *meet* interactions. From about 60,000 frames with 25fps rate, about 25.000 are manually augmented with bounding boxes and the above behaviour labels with ViPER [195].

IXMAS [42] was designed for view-invariant human action recognition, it contains thirteen common motions performed by multiple actors. This dataset is the most similar to HumanEva, having calibrated views from five cameras, background training data, and several behaviour sequences, but it provides no 3D positional ground truth.

On the other hand, CMU-MOCAP (2004–2007) is a large motion capture dataset with 6 categories and 23 subcategories of actions, performed by more than 100 subjects, recorded both as 3D articulated humanoid model (in TVD, C3D, AMC-ASF formats) and video (MPG, AVI) formats. Unfortunately the dataset is aimed at computer animation, and the poor quality video is insufficient for tracking purposes.

The Actions as Space-Time Shapes Dataset [46] of The Weizmann Institute of Science has walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, skip actions, for use in a tracker-less behaviour recognition technique.

The KTH Action database [40] contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. Since the sequences are staged, against homogeneous backgrounds and with a single view, this dataset is limited in the generality of the tracking and recognition.

The CASSANDRA [80] dataset is an audio- and video-sensing dataset for aggressive behaviour detection. The database is not publicly available yet.

## 2.7 Performance evaluation

Given suitable data for evaluation, it needs to be considered what is good performance, for detection, for tracking and for behavioural analysis. The largest cost across the life-cycle of a ship, aircraft or system is manpower [196]. Reducing the overall cost of a system requires

a reduction in the number of humans maintaining it, however easy maintenance requires straightforward performance evaluation and monitoring with well defined performance criteria.

Heckenberg [197] identifies several evaluation methods for human tracking systems, based on quality of the image overlay with the recovered model; the tracking duration; the plot of the tracked parameters for verifying temporal smoothness; comparison of a plot with a reference motion (*i.e.* generated by kinematic studies); verbal evaluation; direct algorithm comparison; ground truth (world or simulated data or manual) and parameter characterisation.

Other metrics [198] include the track accuracy, continuity, total time of success, track fragmentation, temporal fragmentation, identity changes, latency of detection (delay in tracking), track matching error, specific event retrieval performance. Further possibilities are result oriented metric account of the time the operator saves, the familiarity of the proposed method with the current controls, and the effectiveness of the possible hypothesis presentation to the operator.

Heckenberg insists on the importance of synthetic data. This is justified and desired for articulated human tracking, since motion capture systems with markers are elaborate, and depend on the varying human skeleton. Marker placement on the joints is impossible since they are covered by flesh. Therefore the resulting ground truth is an estimation only of the true joint positions.

For behaviour detection in the first four i-LIDS scenarios the event detection rate with $F_\alpha$ score is used [175]:

$$F_\alpha = \frac{(\alpha+1)R\,P}{R+\alpha P}, \tag{2.8}$$

where $TP$, true positives and $FN$, false negatives define the recall, detection rate, and the precision, authenticity of the detection:

$$R = \frac{TP}{TP+FN}, \tag{2.9}$$

$$P = \frac{TP}{TP+FP}, \tag{2.10}$$

with the recall bias $\alpha$ ranging from 0.35 to 75 depending on the scenario.

For the not yet released multi-view tracking i-LIDS dataset, the quality of the tracking

is evaluated over time with the $F_1$ score:

$$F_1 = \frac{2R\,P}{R+P}, \tag{2.11}$$

where precision and recall are:

$$P = \frac{n_{overlap}}{n_{tracked}} \text{ and} \tag{2.12}$$

$$R = \frac{n_{overlap}}{n_{gt}}, \tag{2.13}$$

with $n_{overlap}$ overlapping pixels and $n_{tracked}$ total tracked pixels and $n_{gt}$ total ground truth pixels.

Similar to the i-LIDS metrics, the PETS metrics [199] are Negative Rate, Misclassification Penalty, Rate of Misclassification, Weighted Quality Measure. The on-line evaluation of the results[3] allows objective comparison of the blob tracking algorithms. For alarm and warning signalling in PETS scenarios, true and false positives, and temporal and spatial accuracy of the true positives are used.

Further, apart from surveillance-type blob tracking, in articulated motion tracking, Bălan *et al.* [141] propose the 3D and 2D joint position based error metrics included by Sigal *et al.* [194] in the HumanEva evaluation methodologies. The difference between two body configuration is the difference of the joint locations $\mathrm{X}_m$ and $\hat{\mathrm{X}}_m$ results in the absolute error:

$$D_a(\mathcal{X}, \hat{\mathcal{X}}, \hat{\Delta}) = \frac{\sum_{m=1}^{M} \hat{\delta}_m \|\mathrm{X}_m - \hat{\mathrm{X}}_m\|}{\sum_{i=1}^{M} \hat{\delta}_i}. \tag{2.14}$$

$\hat{\Delta}$ is a binary vector that selects the joint positions $\mathrm{X}_m$ and $\hat{\mathrm{X}}_m$ used by the metric from the joint position vector $\mathcal{X}$ respectively $\hat{\mathcal{X}}$.

Alternatively, the relative error

$$D_r(\mathcal{X}, \hat{\mathcal{X}}, \hat{\Delta}) = \frac{\sum_{m=1}^{M} \hat{\delta}_m \|(\mathrm{X}_m - \mathrm{X}_{torso}) - (\hat{\mathrm{X}}_m - \hat{\mathrm{X}}_{torso})\|}{\sum_{i=1}^{M} \hat{\delta}_i}, \tag{2.15}$$

removes the global position error of the body, by subtracting the torso position from each joint position. It measures the differences of the two poses (*i.e.* joint angles) and evaluates only the reconstruction of the pose [98, 147].

The metrics of the full video sequence are the mean and variance of the absolute or

[3] http://www.cvg.cs.rdg.ac.uk/cgi-bin/PETSMETRICS/page.cgi?home

the relative errors of the selected joints over $t$ frames:

$$\mu_{seq} = \underset{i\in\{1,t\}}{\mathcal{E}} D(\mathcal{X}_i, \hat{\mathcal{X}}_i, \hat{\Delta}), \tag{2.16}$$

and

$$\sigma_{seq} = \sqrt{\underset{i\in\{1,t\}}{\mathcal{E}} [D(\mathcal{X}_i, \hat{\mathcal{X}}_i, \hat{\Delta}) - \mu_{seq}]}. \tag{2.17}$$

This single mean, $\mu_{seq}$, allows simple comparison of different tracking algorithms: in [114], it varies from 100 to 600cm, in particular around 150–200cm, while it is 35–60mm in [141] and 31.36mm in [147].

For behavioural analysis, the recognised actions have to match the ground truth (*e.g.* labelled by an expert). Evaluation of multi-class problem is difficult, since confusion matrices plotting detected classes against the true classes show the possible diffusion in between classes, however comparing matrices is difficult and an objective scalar metric assess them better. Rees*et al.* [200] extend *Receiver Operating Characteristics* (ROC) analyses for multi-class problem for finding the optimal operating point of the system, however the ROC still needs visual evaluation. As an alternative, the accuracy [201], or correct classification percentage, of a confusion matrix C defined as

$$\zeta = \frac{\sum_i \mathrm{C}_{i,i}}{\sum_{i,j} \mathrm{C}_{i,j}}, \tag{2.18}$$

provides a scalar metric for the multi-class classification.

If algorithms are designed for a specific task or environment (*e.g.* without shadows) they cannot be compared objectively. Therefore well-specified datasets, such as PETS or HumanEva are required, with defined training and testing sequences, ground truth and evaluation metrics. Such datasets are costly, and the acquisition is a standalone project. Subsystem evaluation is important to verify parts of the system. However, if the system is aiming for a specific goal, that has to be checked against that goal, on the final output. This, for behavioural systems is the quality of the recognition.

## 2.8 Summary

Visual analysis of humans is a very active domain, therefore this chapter was devoted to multiple aspects: first to the psychological background of the behavioural analysis, then to

prior knowledge and image measurements for human tracking. Further, several complete behavioural systems were presented, together with the necessary datasets and evaluation techniques for training and validation. From other aspects, the human tracking related to behavioural analyses, the articulated motion reconstruction and human-computer interaction are reviewed by Gavrila [202], Hu *et al.* [203], Moeslund and Granum [204], Moeslund *et al.* [205] and by Sidenbladh [197].

At the beginning of the chapter, behavioural analyses were classified into methods that do or do not use tracking. It was concluded that an intermediate model, recovered by the tracking process, is required for generality. This is the approach taken in the later chapters, hence chapter 4 concerns the understanding of behaviour only from exact model parameters, chapter 6 recovers these from videos sequences, and chapter 5 deals with the fusion of the two.

Tracking is most flexible in a stochastic framework for recovering model parameters, since this allows probabilistic description of output parameters and late decisions in behavioural analysis. Specifically, the particle filter works with multiple concurrent hypotheses, and resembles human cognitive processes in being a generative approach. However, basic PF algorithms have to be modified, in chapter 5, for human tracking.

In comparison to 2D or blob models, 3D articulated models provide the most detailed and realistic description of the target, which is mandatory if the behaviour to be analysed is specialised and detailed. 3D models allow implicit occlusion and self-occlusion reasoning. The 3D model defined in chapter 3 is tracked (chapter 5) and analysed (chapter 4).

Environmental knowledge is easy, but valuable information. Although the full perspective model is computationally more expensive than the orthographic model, it allows depth dependent, accurate image formation. For 3D models, the camera model and calibration are imperative and are discussed in chapter 5. Static camera setup allows background modelling that greatly simplifies the observation data. The model chosen for this thesis is the widely applied Stauffer's mixture of $K$ Gaussian distributions, described earlier in this chapter.

Knowledge about human motion is important in tracking, as it predicts the next configuration and reduces the search space. Therefore, chapter 4 analyses several motion models, learnt from the training data. However motion, as was seen earlier in psychological experiments, is the basic block for behavioural analysis. Hence, chapter 4 also links the motion model to behaviour that is not only global, general actions, but are fine-scaled,

defined by one or more limbs.

There is no recipe for the best likelihood of an image, however several alternatives were presented. Chapter 3, apart from the human model, defines the a multi-modal and multi-part observation model, embodied by a mixture of likelihoods, incorporated into the PF in chapter 5.

This chapter has also introduced the HumanEva, CAVIAR and i-LIDS datasets and the HumanEva evaluation metric. The HumanEva provides both image and ground truth data for articulated tracking, thus it is the main testing data used in the thesis. CAVIAR and i-LIDS are realistic test scenarios, however their augmentation and detail are limited, and therefore they can be used only for visual evaluation. The available testing data is limited, in the sense that behaviours are global activities only, without a fine-grained description.

The majority of tracking applications treat humans as blobs, limiting the range of behaviours. Therefore the rest of this thesis focuses *equally* on tracking and behaviour, aiming for the generality in both.

Problems identified, chapter 3 focuses on the static human and observation models that in chapter 4 allow definition of a motion and behavioural model and in chapter 5, capitalising on the learnt motion model, recovers with a particle filter the effective articulated motion from video sequences. Finlay the result of the combined tracking and behavioural analysis is presented in chapter 6.

# Chapter 3

# Models for human tracking

Chapter 2 emphasised the importance of models that provide hard coded or learnt priors. This chapter defines several in-built assumptions and priors that ground the analysis of the later chapters. It introduces the articulated 3D human model, and the related scene and observation models. Based on these, learnt motion and behavioural models will be added in the next chapter and all models will contribute to the tracking.

The tracking method considered in the thesis is the particle filter, a generative approach (section 2.2.2) that requires only the easier, direct 3D to 2D projection, and for this the 3D environment and camera models are considered next. Then, in this space, the 3D articulated human model, and the observation model that connects the 3D model with the image are defined.

## 3.1 The three-dimensional space and the camera model

The two common camera models, orthographic and perspective, were introduced in section 2.3.3. If calibrated, the perspective camera model provides accurate measurements for an arbitrary scene. The perspective transformation present in surveillance situations (*e.g.* CAVIAR, i-LIDS), motivates also this model. As well as using and adapting the existing mathematical model, this section calibrates the perspective model for the case when no prerequisites were made for calibration during the video acquisition.

Hartley and Zisserman [206, pp.158–164] summarises the Tsai [174] camera model for pinhole cameras; the relations between image and world points are represented as homogeneous equations. The perspective camera maps a 3D point from the *World Coordinate*

*System* (WCS) into a 2D point in the *Image Coordinate System* (ICS). In this thesis, a right handed coordinate system is considered, unless otherwise specified.

The transformation between WCS to ICS is performed through the *Camera Coordinate System* (CCS). The relations between WCS, CCS and ICS from figure 3.1 are expanded next.

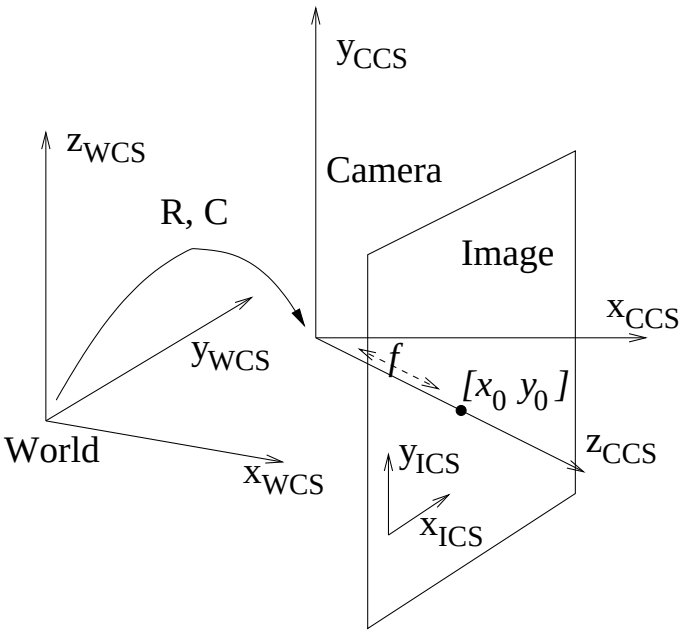



Figure 3.1: The relation between world, camera and image coordinate systems. The affine transformation with rotation R and translation C define the CCS in the WCS, while the focal length $f$ and principal point $[x_0\ y_0]^T$ give the ICS in the camera coordinate frame.

The 3D point homogeneous coordinate $\mathrm{X} = [X\ Y\ Z\ 1]^T$ in the CCS has the coordinates $\mathrm{x}_c = [x_c\ y_c\ z_c]^T$ defined by the equation:

$$\mathrm{x}_c = \mathrm{R}\,[\,\mathrm{I}\,|\,-\mathrm{C}]\,\mathrm{X}. \tag{3.1}$$

The camera extrinsic parameters, are R, the CCS orientation and C, the camera centre in the WCS. R is a $3 \times 3$ rotation transform between WCS and CCS, represented concisely as the product of three rotation matrices around x, y and z axis with pan ($\alpha$), tilt ($\beta$) and yaw ($\gamma$) angles:

$$\mathrm{R}(\alpha, \beta, \gamma) = \mathrm{R}_x(\alpha)\mathrm{R}_y(\beta)\mathrm{R}_z(\gamma), \tag{3.2}$$

Expanding:

$$\mathrm{R}(\alpha,\beta,\gamma) = \begin{bmatrix} \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & -\cos\beta\sin\gamma & -\sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma \\ \cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & \cos\beta\cos\gamma & -\sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma \\ \sin\alpha\cos\beta & \sin\beta & \cos\alpha\cos\beta \end{bmatrix}. \tag{3.3}$$

The projected homogeneous point coordinate in the ICS is

$$\mathrm{x} = \mathrm{K}\,\hat{\mathrm{x}}_c, \tag{3.4}$$

where K is the camera intrinsic parameter matrix

$$\mathrm{K} = \begin{bmatrix} a_x & s & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.5}$$

with the skew $s \neq 0$ for non perpendicular sensor axis; the camera principal point $[x_0\, y_0]^T$ in ICS; and, $a_x$ and $a_y$, the vertical and horizontal focal lengths in terms of image pixels. For general purpose cameras $s = 0$ and $a_x = a_y = f$. K is constant and specific to the camera optics.

For ideal pinhole cameras

$$\hat{\mathrm{x}}_c = \mathrm{x}_c, \tag{3.6}$$

however for real, non-pinhole lenses there is a non-linear component introduced by the lens distortion

$$\hat{\mathrm{x}}_c = \mathtt{L}_r(r)\,\tilde{\mathrm{x}}_c + \mathtt{L}_t(r, \tilde{\mathrm{x}}_c), \tag{3.7}$$

dependent on the normalised coordinate $\tilde{\mathrm{x}} = [\tilde{x}_c\, \tilde{y}_c\, 1]^T$ with

$$\tilde{x}_c = \frac{x_c}{z_c} \text{ and } \tilde{y}_c = \frac{y_c}{z_c}, \tag{3.8}$$

and on distance from camera centre

$$r = \sqrt{\tilde{x}_c^2 + \tilde{y}_c^2}. \tag{3.9}$$

$\mathtt{L}_r$ is the radial, while $\mathtt{L}_t$ is the tangential distortions:

$$\mathtt{L}_r(r) = \begin{bmatrix} 1+\kappa_1 r^2+\kappa_2 r^4+\kappa_5 r^6 & 0 & 0 \\ 0 & 1+\kappa_1 r^2+\kappa_2 r^4+\kappa_5 r^6 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and} \tag{3.10}$$

$$\mathtt{L}_t(r, \tilde{\mathrm{x}}_c) = \begin{bmatrix} 2\kappa_3 \tilde{x}_c \tilde{y}_c + \kappa_4(r^2+2\tilde{x}_c^2) \\ \kappa_3(r^2+2\tilde{y}_c^2) + 2\kappa_4 \tilde{x}_c \tilde{y}_c \\ 0 \end{bmatrix}, \tag{3.11}$$

where $\kappa$ is the six degree radial distortion vector.

Without radial distortion, combined equations (3.1), (3.4) and (3.6) result in

$$\mathrm{x} = \mathrm{Q}\,\mathrm{X}, \tag{3.12}$$

where Q is the camera projection matrix

$$\mathrm{Q} = \mathrm{K}\,\mathrm{R}\,[\,\mathrm{I}\,|\,-\mathrm{C}\,]. \tag{3.13}$$

For the general case, if the lens is distorting, equation (3.12) becomes

$$\mathrm{x} = \mathtt{Q}(\mathrm{X}), \tag{3.14}$$

with $\mathtt{Q}$ a function that includes the nonlinear transformation of equation (3.7). For notation simplicity in this thesis we use the linear form of equation (3.13), replaceable, if distortion is relevant, with equation (3.14).

For a camera, the intrinsic parameters K are fixed and known. The calibration is the recovery of the projection matrix Q by the the means of the extrinsic parameters R and C. If calibrated, real world points are mapped uniquely to single image points, however one 2D point corresponds to a line in the higher dimensional 3D space.

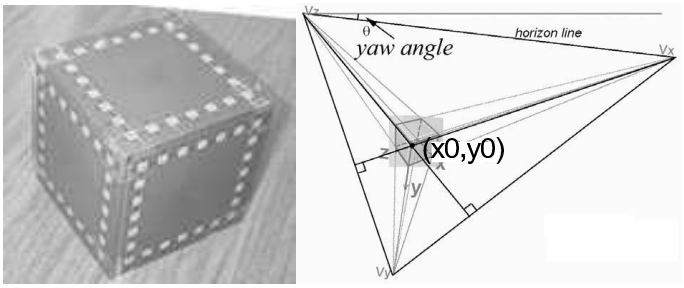
Only calibrated images provide measures of real world distances and angles. Ideally, calibration parameters are recovered and stored in advance of the main processing algorithm. When the scene is known, calibration can be performed by specifying 3D to 2D point correspondences, explained in section 3.1.2 in the context of an unknown scene.

In case of the HumanEva dataset, the cameras are calibrated, and the parameters

R, C, $\kappa$ and K are available. Unfortunately, other datasets (*e.g.* CAVIAR, i-LIDS) are uncalibrated. For these, the calibration matrix has to be recovered without available real word measurements. Next, two calibration methods, adapted for this case, are reviewed.

### 3.1.1 Calibration with vanishing points

Criminisi *et al.* [207] present a virtual model of a scene built from a single image, while [208] shows how measurements are taken from an initially uncalibrated image. For both, only images that need to be calibrated are used. The calibration for a zero skew, equal horizontal and vertical focal length camera, positioned along the y axis above the origin ($\mathrm{C} = [0\, H_c\, 0]^T$) is as follows [209–211]:




(a) (b)

Figure 3.2: Definition of vanishing points with a cube from Lv *et al.* [211]. The parallel lines of an image, the edges of a cube in (a), intersect in vanishing points $V_x$, $V_y$ and $V_z$, as shown in (b). These in pairs define the vanishing lines. The vanishing line parallel to the ground plane is the horizon line.

1. The vanishing points ($V_x, V_y$ and $V_z$) are the intersection of three pairs of parallel lines in 3D with the WCS axis (figure 3.2). Note, that projection of lines parallel in 3D results in intersecting lines in the image plane.

2. The principal point $[x_0\, y_0]^T$ is the orthocentre of the triangle defined by the vanishing points.

3. The yaw angle $\gamma$ is the angle between the horizon line ($(V_x\, V_z)$ vanishing line) and the horizontal image axis.

4. The focal length, the pan and the tilt angles are

$$f = \sqrt{-(y_{V_x} - y_0)(y_{V_y} - y_0)}, \tag{3.15}$$

$$\beta = \arctan\left(\frac{y_0 - y_{V_x}}{f}\right), \tag{3.16}$$

$$\alpha = \arctan\left((x_{V_x} - x_0)\frac{\cos\beta}{f}\right), \tag{3.17}$$

5. Camera centre position $H_c$ is computed from the real length of a vertical segment [211].

Automated calibration involves step 1 to 4, while step 5 needs a simple human intervention assigning the length of a segment. Bose and Grimson [212] uses affine and metric rectification, based on the path of tracked moving objects. They assume that the centroid of the moving object lies on the ground plane, true when the camera is well above the moving object.

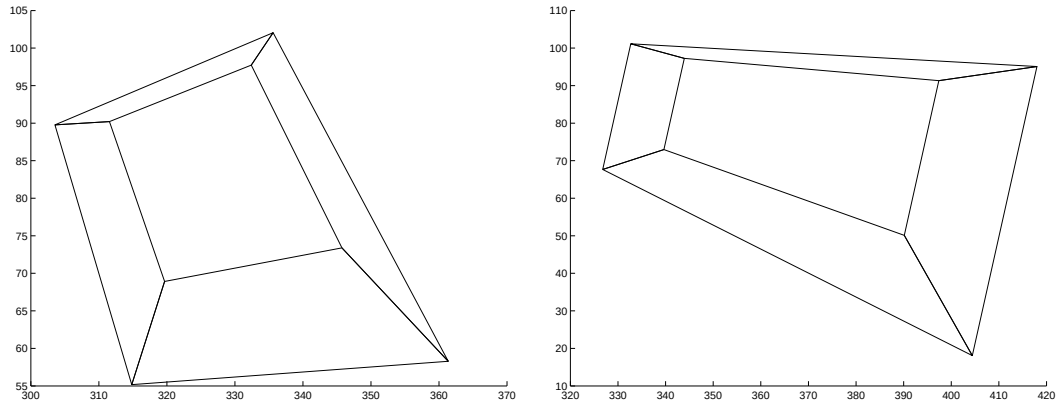
**Example**

This example evaluates the vanishing point based calibration on two synthetic images using three pairs of parallel lines, orthogonal in pairs, applying the above steps 1 to 4. The points defining the lines are given first with their exact, then, selected manually, with the approximated coordinates.

The tests are performed on two synthetic images, both with a projected 3D cube, with edges parallel with the WCS axis, and with the two diagonal vertexes in $[10\ \ 10\ \ 10]^T$ and $[110\ \ 110\ \ 110]^T$. The first projection has focal length $f = 50$, principal point $[300\ 100]^T$, zero skew; camera frame rotation with $\alpha = \frac{5}{18}\pi$ pan, $\beta = \frac{1}{3}\pi$ tilt, $\gamma = \frac{1}{10}\pi$ yaw angles and camera centre at $T = [-150\ \ 80\ \ 90]^T$ (figure 3.3(a)). The second projection is less perspective, with angles $\alpha = \frac{1}{2}\pi$, $\beta = \frac{3}{16}\pi$ and $\gamma = \frac{1}{18}\pi$ and with all other parameters unchanged (figure 3.3(b)).

The accuracy of the calibration is checked comparing the real principal point and the focal length against the recovered values from five different sets, each with three pairs of parallel edges. First, the edges are given by the exact vertex coordinates. In this case the *exact* mathematical expression of the supporting three pairs of parallel lines is known. For figure 3.3(a), is results a stable focal length $f = 50$ and principal point $[302\ \ 100]^T$. However, for the second projection, the recovered parameters, table 3.1 are unstable, with large variance of the principal point and the focal length.

In the second test, computations are identical, however the parallel edge pairs are marked *manually*. The recovered principal points and focal lengths for the two projections




(a) (b)

Figure 3.3: Test cube with high (a) and low (b) perspective transformation. In the right figure the most vertical edges of the cube are near-parallel, therefore the vanishing point is far out of the image frame, while for the left image all vanishing points are closer to the image frame, thus more accurately defined.

| Principal point $[x_0 \;\; y_0]$ | Focal length $f$ |
|---|---|
| [6439 397] | $\infty$ |
| [287 102] | 41 |
| [298 100] | 48 |
| [319 97] | 54 |
| [321 97] | 54 |

Table 3.1: Recovered principal point and focal length for the *second* projection from five *exact* edge sets

are shown in tables 3.2 and 3.3. These parameters are less stable compared to the exact edges. The errors originate in inaccuracies of the manually marking, an error of a few pixels can generate large changes in the intersections of the lines, and therefore in the position of the vanishing point.

| Principal point $[x_0 \;\; y_0]$ | Focal length $f$ |
|---|---|
| [291 60] | 60 |
| [290 54] | 54 |
| [283 79] | 79 |
| [288 62] | 62 |
| [289 61] | 61 |

Table 3.2: Recovered principal point and focal length for the *first* projection from five *manually* marked edge sets.

| Principal point $[x_0 \;\; y_0]$ | Focal length $f$ |
|---|---|
| [310 99] | 55 |
| [345 93] | 50 |
| [4976 651] | $\infty$ |
| [4666 718] | 36 |
| [600 89] | $\infty$ |

Table 3.3: Recovered principal point and focal length for the *second* projection from five *manually* marked edge sets.

This instability matches the remarks of Trucco and Verri [210, p.132] that, if vanishing points are not close to the image centre, then small inaccuracies in the location defining the lines result in large error of the vanishing points, that compromises the principal point and therefore equations (3.16) and (3.17).

To conclude, calibration with vanishing lines is highly dependent on how accurately the perspective effect on the image can be measured, and on the accuracy of the manual markings. With reduced perspective effect, even a small inaccuracy in the specified parallel lines results in incorrect calibration. The method is accurate only if no WCS axis is parallel or nearly parallel with the image plane, and the defining lines of the vanishing points can be specified in the image with high accuracy.

### 3.1.2 Calibration with point correspondences

Willson's implementation [213] of Tsai's algorithm [174] is frequently used for calibration (*e.g.* in PETS2006 dataset). With at least five coplanar points, or with seven non-coplanar points both in WCS and ICS, and with a set of rough camera intrinsic parameters, Tsai computes the 17 camera calibration parameters. These are 6 external parameters (3 translations, $\mathrm{C} = [C_x\, C_y\, C_z]^T$, 3 Euler angles, $\alpha$, $\beta$ and $\gamma$ for R), 5 internal parameters (principal point $[x_0 \;\; y_0]$, the unique focal length, $f$, the first order radial distortion, $\kappa_1$, and the skew $s$) and 6 camera related intrinsic constants (number of sensors $Nfx$, and pixels, $Ncx$, in frame grabber; sensor dimensions $dx, dy$, and frame grabber resolutions $dpx, dpy$).

Tsai uses left-handed coordinate systems, therefore compared to equation (3.3) the expression for R is:

$$\mathrm{R} = \begin{bmatrix} \cos\beta\cos\gamma & \cos\gamma\sin\alpha\sin\beta - \cos\alpha\sin\gamma & \sin\alpha\sin\gamma \;+ \cos\alpha\cos\gamma\sin\beta \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \cos\gamma\sin\alpha \\ -\sin\gamma & \cos\beta\sin\alpha & \cos\alpha\cos\beta \end{bmatrix}. \quad (3.18)$$

If the radial distortion is ignored, the recovered parameters with equations (3.3), (3.5) and (3.13) lead to the projection matrix Q.

The method calibrates even without access to measurements of the recorded scene, if 3D coordinates of manually selected pixels in the uncalibrated image can be guessed. The calibration is not perfect, but it is accurate for most tracking applications.

#### Example

This example manually calibrates the corridor scene of the CAVIAR video sequence, employing a set of 5–7 correspondences of 2D and 3D points.

A Matlab application was built for manually assigning marked image points to world points entered as 3D coordinates. The application visualises the points both in the ICS and the WCS (figure 3.4). The WCS origin, axis and scale are arbitrary, allowing the operator the most appropriate convention. In order to keep the scale accurate, the ground plane points and the common human height are useful clues. The application allows addition, deletion, saving or loading of points. After a set of points is fully specified, the 3D and 2D coordinates are passed to Willson's algorithm [213], which generates the calibration

parameters for the Q projection matrix. The non-linear lens distortion of the camera was ignored in this process.

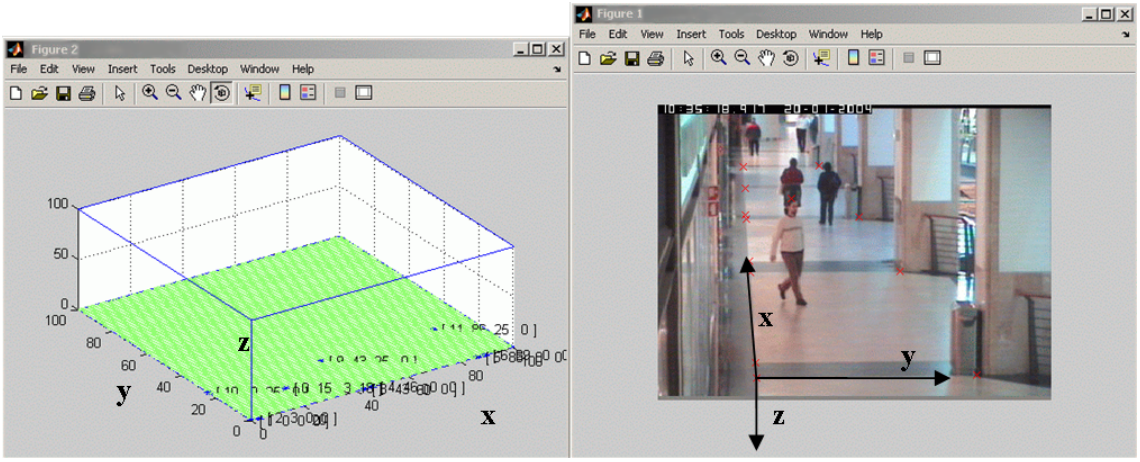



Figure 3.4: Calibration tool of an uncalibrated image. The user selects points on the image and enters their 3D coordinates. The tools shows the points in WCS (left) and ICS (right).

Both coplanar and non-coplanar point sets were tested for calibration. Non-coplanar calibration is less accurate, however a set of eight coplanar points on the ground plane with positions estimated from the regular floor pattern gives good results. Testing transformation with points on the ground plane from 3D to 2D and from 2D to 3D shows accuracy, however points not on ground have higher errors. Although not perfect, it is appropriate for tracking, since precise localisation and the size of the object is not essential, and the error is constant. More accurate calibration could be possible with an extended version of the tool that computes the projection matrix concurrent with newly added points, and visualises continuously the positive of adverse effect on the re-estimated other points. It was learnt that choosing points that present perspective distortion and do not form planes parallel with the image is essential.

The calibration quality was checked both by 3D points projected in 2D, and by 2D points marked on the image with a given depth recovering the 3D position. Both tests are highly empirical since no real 3D position is known. The results are *visually* correct.

Further, the 3D position of the walking figures tracked with the Reading Tracker (section 2.5) was visualised. The tracked contour has 32 control points, including head, feet, elbows and hips of the body. Back-projecting these special points to 3D, and assuming feet on the ground and a person with zero depth, then a simple 3D rendering can visualise the scene. Two frames and their reconstructions are shown in figure 3.5. It is verified

visually that the calibration is correct for the whole sequence.

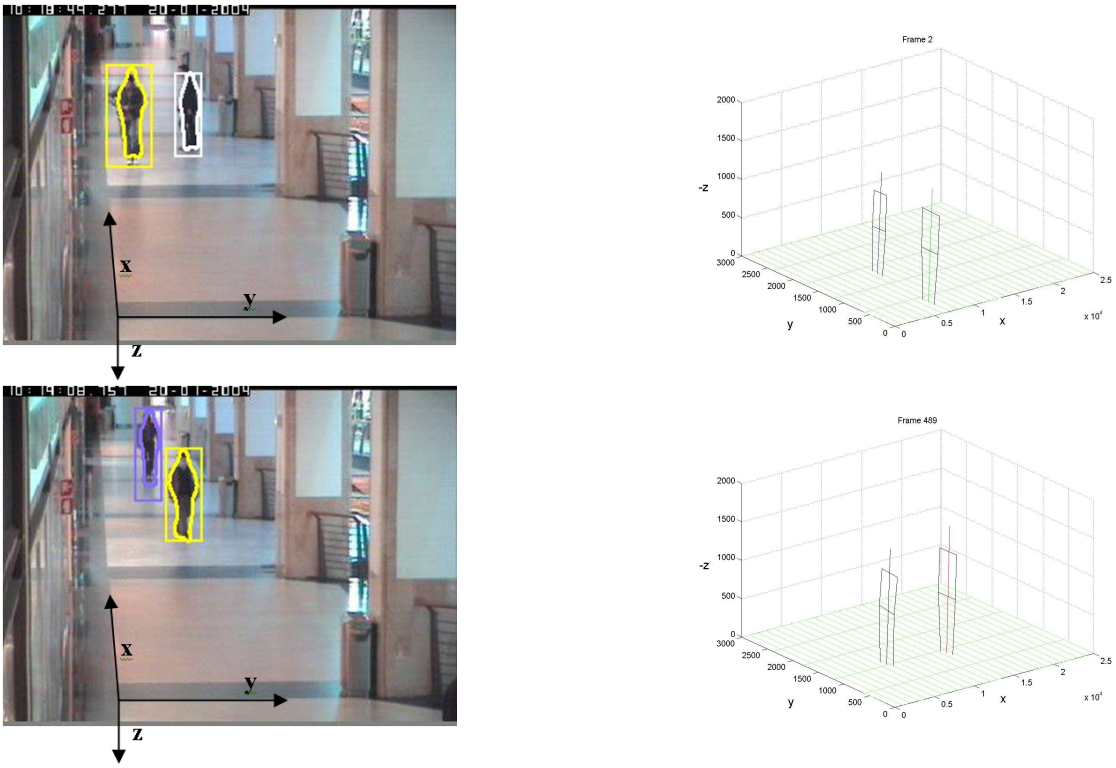



Figure 3.5: Calibration test with the Reading Tracker. Reading Tracker results on the CAVIAR OneStopEnter2cor sequence and 3D reconstructions of frame no. 2 and 489 verify the manual calibration. In the first frame the tracked persons have the same depth, while the second 3D model shows a displacement.

### 3.1.3 Conclusions on calibration

Vanishing lines are inappropriate to calibrate tracking scenes. The manual intervention of specifying lines is inaccurate due to unavoidable marking errors. For light perspective transformation, usually the vertical axis generates nearly parallel calibration lines with the image plane. This makes impossible the calibration [210].

The Tsai-Willson approach has a similar problem if the calibration points are not on the ground plane. and points at different heights cause large errors. Fortunately, ground plane points permit good calibration of the scene.

### 3.1.4 Test sequence calibration

The CAVIAR and i-LIDS datasets are not calibrated. For the purpose of 3D tracking, they were calibrated with point correspondences developed earlier in section 3.1.2. For re-

producibility, the point correspondences along and the computed calibration are described next.

### The CAVIAR sequences

ICS points 1–6, figure 3.6(a) are manually selected with the developed Matlab application. For each $x_i$ ICS coordinate an $X_i$ WCS point is assigned as table 3.4 shows. The guesses are based on the four corridor points and 4 frontal points provided by the CAVIAR dataset for ground plane homography computation. To conform the CAVIAR convention, the x and y axis are rotated, while the z axis, in contrast with previous calibration, figures 3.4 and 3.5, is radiating out from the ground plane. The 2D and 3D coordinate pairs, feed into Wilson's application and generate the calibration values from table 3.7, second column. For calibration testing, the 3D points (1–6) are reprojected onto the image, figure 3.6(b). Visually, as well as the numerical values, $\check{x}_i$, from table 3.4 are near to their initial position, verified also by the low mean and standard deviation of distances from initial values. Points 7–8, the legs and head of the walking man, attest a good localisation and that the z axis is calibrated. A possible problem, however is the 1700mm height of the human, though without valid measurements it is reasonable to accept this low value.


(a) Calibration points ($x_i$) (b) Test points ($\check{x}_i$)

Figure 3.6: CAVIAR corridor view calibration.

A similar process was repeated for the frontal view, using where was possible, the same points (figure 3.5). Table 3.5 gives the 3D to 2D correspondences and the back-projected calibration points. The obtained calibration is provided by the third column of table 3.7. Points 7–8, in both CAVIAR views lay on the walking human that verifies the calibration.

| No. | $X_i$[mm] | | | $x_i$[pixel] | | $\check{x}_i$[pixel] | |
|---|---|---|---|---|---|---|---|
| | x | y | z | x | y | x | y |
| 1 | 0 | 9750 | 0 | 91 | 163 | 91.8 | 162.9 |
| 2 | 2900 | 9750 | 0 | 241 | 163 | 236.3 | 161.1 |
| 3 | 0 | -1100 | 0 | 98 | 266 | 97.6 | 266.5 |
| 4 | 2900 | -1100 | 0 | 322 | 265 | 321.5 | 264.5 |
| 5 | 0 | 0 | 0 | 97 | 251 | 97.0 | 251.0 |
| 6 | 3820 | 8780 | 0 | 285 | 166 | 288.6 | 166.9 |
| 7 | 1000 | 5100 | 0 | NA | NA | 153.4 | 195.3 |
| 8 | 1000 | 5100 | 1700 | NA | NA | 152.9 | 97.0 |

Table 3.4: Calibration and test points for the CAVIAR corridor scene. Mean error of the calibration points is 1.85606 with standard deviation 2.06155.

| No. | $X_i$[mm] | | | $x_i$[pixel] | | $\check{x}_i$[pixel] | |
|---|---|---|---|---|---|---|---|
| | x | y | z | x | y | x | y |
| 1 | 0 | 0 | 0 | 60 | 153 | 59.0 | 153.1 |
| 2 | 0 | 9750 | 0 | 359 | 153 | 362.8 | 153.2 |
| 3 | 3820 | 980 | 0 | 50 | 201 | 51.2 | 200.6 |
| 4 | 3820 | 8780 | 0 | 367 | 200 | 368.3 | 200.4 |
| 5 | 0 | -1100 | 0 | 27 | 153 | 30.5 | 152.8 |
| 6 | 2900 | 0 | 0 | 28 | 186 | 29.7 | 185.5 |
| 7 | 1000 | 5100 | 0 | NA | NA | 221.3 | 165.2 |
| 8 | 1000 | 5100 | 1700 | NA | NA | 221.4 | 104.5 |

Table 3.5: Calibration and test points for the CAVIAR frontal scene. Mean error of the calibration points is 2.10479 with standard deviation 1.21089.


(a) Calibration points ($x_i$) (b) Test points ($\check{x}_i$)

Figure 3.7: CAVIAR frontal view calibration.

| No. | $X_i$[mm] | | | $x_i$[pixel] | | $\check{x}_i$[pixel] | |
|---|---|---|---|---|---|---|---|
| | x | y | z | x | y | x | y |
| 1 | 0 | 0 | 0 | 93 | 509 | 93.6 | 508.7 |
| 2 | 580 | 0 | 0 | 176 | 476 | 175.1 | 476.4 |
| 3 | 1160 | 0 | 0 | 247 | 448 | 246.7 | 448.0 |
| 4 | 2900 | 0 | 0 | 417 | 381 | 417.5 | 380.2 |
| 5 | 0 | 630 | 0 | 39 | 476 | 38.8 | 476.2 |
| 6 | 1740 | 1260 | 0 | 199 | 381 | 200.0 | 380.4 |
| 7 | 3480 | 3150 | 0 | 223 | 294 | 222.6 | 293.8 |
| 8 | 4640 | 1260 | 0 | 435 | 303 | 435.2 | 304.6 |

Table 3.6: Calibration and test points for the i-LIDS scene. Mean error of the calibration points is 0.810965 with standard deviation 0.465765.

#### i-LIDS sequences calibration

The calibration of i-LIDS is identical, however without exact available scene measurements the main assumption used is the approximate size of a floor tile, being $58 \times 63$ cm (approx. $24 \times 23$ inches). The calibration and test points from table 3.6 are shown in figure 3.8, while calibration parameters are presented in column four of table 3.7.

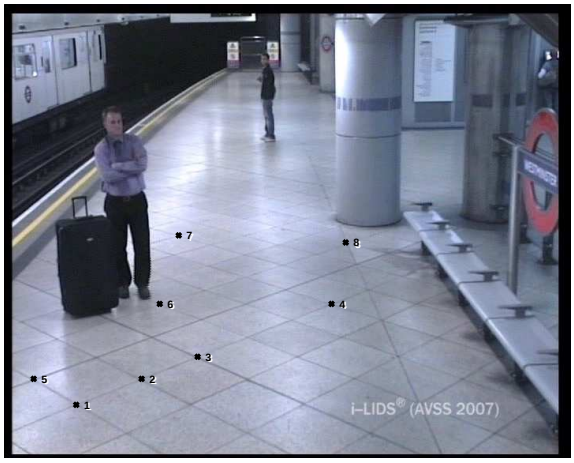



Figure 3.8: i-LIDS calibration.

## 3.2 The Articulated Hierarchical Human Model

It was concluded in section 2.8 that for complex behaviours and complex (*i.e.* 3D) scenes the human model has to be 3D and articulated. Thus, their costs are

- complex tracking algorithms and longer processing time,
- and less stable tracking of the smaller parts.

| Parameter | CAVIAR corridor | CAVIAR frontal | i-LIDS |
|---|---|---|---|
| $f[mm]$ | 1574.8331 | 504.1752 | 1000.5121 |
| $\kappa[1/mm^2]$ | -1.456979e-02 | 4.167905e-02 | 6.327647e-05 |
| $C_x[mm]$ | -1299.228962 | -4239.907034 | -1553.750971 |
| $C_y[mm]$ | 1462.415893 | 291.031975 | 1287.274442 |
| $C_z[mm]$ | 22194.862720 | 14967.979252 | 5830.475932 |
| $\alpha[^\circ]$ | 96.440553 | -169.859950 | 111.340395 |
| $\beta[^\circ]$ | -4.058110 | 73.112182 | -44.085914 |
| $\gamma[^\circ]$ | -0.570705 | 99.740466 | -15.717844 |
| $s$ | 1 | 1 | 1 |
| $x_0[pixels]$ | 192 | 192 | 360 |
| $y_0[pixels]$ | 144 | 144 | 288 |
| $Ncx[sel]$ | 384 | 384 | 768 |
| $Nfx[pixels]$ | 384 | 384 | 768 |
| $dx[mm/sel]$ | 0.01 | 0.01 | 0.01 |
| $dy[mm/sel]$ | 0.01 | 0.01 | 0.01 |
| $dpx[mm/pixels]$ | 0.01 | 0.01 | 0.01 |
| $dpy[mm/pixels]$ | 0.01 | 0.01 | 0.01 |

Table 3.7: Recovered calibration parameters.

For tracking and behavioural analysis, a good model has to

- allow fast, on-line processing, but has to be detailed;
- keep the number of parameters low, but allow a full range of body configurations.

Therefore the designed *Articulated Hierarchical Human Model* (AHHM) consists of body parts that

- to enhance speed, are modelled with simple geometrical objects
- to keep parameters reduced, are defined relative to a parent body part and have limited scale (hands or fingers are ignored) and are fixed in size.

Further, the AHHM consists of 12 body parts (figure 3.9), each a frustum, with elliptical cross-section. Body parts are defined relative to a hierarchically superior part (table 3.8), a parent limb, or for the torso, relative to the WCS. Hands are not modelled since regularly they are not visible on the targeted on surveillance situations. However, legs are visible and provide important clue about the leg *twist* angle (parameters 19 and 23 in table 3.9).

The radius of the two the elliptical bases, the height of the parts and the joint positions relative to the parent limb are constant. These are either set manually or supplied from

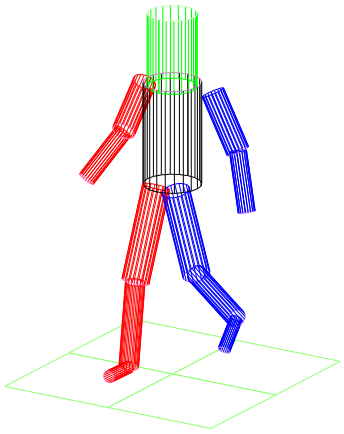


Figure 3.9: A body pose with the AHHM. The colour convention used throughout the thesis in order to disambiguate left and right parts is: right limbs are red, left limbs are blue, trunk is black, head is green.

the ground truth in the first frame. Initialisation of such a model is complex, and in this work we consider it to be known.

### 3.2.1 Parametrisation of the AHHM

Global position and joint angles, 24-parameters (table 3.9), define the location and the pose of the body. To exclude impossible poses, each parameter is restricted to a range prescribed by physical constraints of the human body, given in table 3.10. In order to limit complexity, these do not exclude self intersecting body parts, as this requires joint volumetric analysis of multiple limbs. Although recent work [100,173] shows that physical constraints and laws successfully model interactions between body parts or between the body and the environment, such conditions are also ignored in this thesis.

The constraints that we embed into AHHM are represented for each limb by a range $[0; r^x], r^x \in \mathbb{N}$, chosen for each parameter. A larger $r^x$ allows denser, more detailed representation of the parameter. This was considered for joint angles that have larger variations, as shown in table 3.10.

The scaled 24-parameter vector (table 3.9) is the *Pose Vector* (PV) and describes completely the location and configuration of the AHHM. A single parameter is $p^x$, where $x$ is the parameter index (*i.e.* $p^{17}$ is left leg frontal rise) and $\mathrm{p}^\phi$ with $\phi \subseteq \{1, .., 24\}$ is partition of the PV (*e.g.* the partition $\mathrm{p}^{\{1,2,3\}}$ is the global position of the body).

| Part | Parent | Rotation | | |
|---|---|---|---|---|
| | | $\theta_x$ | $\theta_y$ | $\theta_z$ |
| Torso | World | $-p^5$ | $p^6$ | $p^4$ |
| Head | Torso | $\pi + p^8$ | $\pi$ | $p^7$ |
| L upper arm | Torso | $\pi + p^9$ | $-p^{10}$ | $\pi + p^{11}$ |
| L lower arm | L upper arm | 0 | $-p^{12}$ | 0 |
| R upper arm | Torso | $\pi + p^{13}$ | $p^{14}$ | $\pi - p^{15}$ |
| R lower arm | R upper arm | 0 | $p^{16}$ | 0 |
| L upper leg | Torso | $\pi + p^{17}$ | $-p^{18}$ | $p^{19}$ |
| L lower leg | L upper leg | $-p^{20}$ | 0 | 0 |
| L feet | L lower leg | $\pi/2$ | 0 | $\pi/2$ |
| R upper leg | Torso | $\pi + p^{21}$ | $p^{22}$ | $\pi - p^{23}$ |
| R lower leg | R upper leg | $p^{24}$ | 0 | 0 |
| R feet | R lower leg | $-\pi/2$ | 0 | $-\pi/2$ |

Table 3.8: Limb coordinate systems definitions. Limb coordinate system relative position, and definition of the parameters (joint angles) based rotations.

### 3.2.2 Parameter range constraint

Table 3.9 defined the valid range $[0; r^x]$ of the parameter for a parameter $p^x$. Therefore the range prior, $\pi_r(\mathrm{p}^\phi)$ of the values $x \in \phi$ is defined as:

$$\pi_r(\mathrm{p}^\phi) = \prod_{x \in \phi} e^{-0.5 d(p^x)}, \tag{3.19}$$

with

$$d(p^x) = \begin{cases} 0, & \text{if} \quad 0 \leq p^x \leq r^x \\ -p^x & \text{if} \quad p^x < 0 \\ p^x - r^x & \text{if} \quad p^x > r^x \end{cases} . \tag{3.20}$$

### 3.2.3 The limb coordinate systems

Each body limb of the AHHM is defined relative to its parent limb. Figure 3.10 shows the orientation of *Limb Coordinate Systems* (LCS) in the neutral pose with all rotation parameters of table 3.8 equal $p^x = 0°, x = 4..24$.

The origin of the LCS is the connecting joint to the parent limb. The direction of z axis is along the limb axis, towards the far end of the limb, while rotations of axes are defined by table 3.8 and figure 3.10. The forward rotation of any limb, with the conversions from table 3.8, increases the appropriate joint angle, $p^x$.

A point $\mathrm{X}_{l_1}$ in the $l_1$ limb's LCS has the coordinate $\mathrm{X}_{l_2}$ in the $l_2$ parent LCS, provided

| Parameter no | Name |
|---|---|
| 1 | torso (root) x coordinate |
| 2 | torso (root) y coordinate |
| 3 | torso (root) z coordinate |
| 4 | heading orientation |
| 5 | spine inclination |
| 6 | spine tilt |
| 7 | head orientation |
| 8 | head inclination |
| 9 | left arm frontal rise |
| 10 | left arm side rise |
| 11 | left arm twist |
| 12 | left elbow |
| 13 | right arm frontal rise |
| 14 | right arm side rise |
| 15 | right arm twist |
| 16 | right elbow |
| 17 | left leg frontal rise |
| 18 | left leg side rise |
| 19 | left leg twist |
| 20 | left knee |
| 21 | right leg frontal rise |
| 22 | right leg side rise |
| 23 | right leg twist |
| 24 | right knee |

Table 3.9: The 24 parameters of the pose vector.

by the homogeneous transformation

$$\mathrm{X}_{l_2} = \mathrm{T}_{l1}^{l2}(t, \theta_x, \theta_y, \theta_z)\mathrm{X}_{l_1}. \tag{3.21}$$

The transformation $\mathrm{T}_{l1}^{l2}$, consists of three rotations ($\theta_x$, $\theta_y$ respectively $\theta_z$) and a t translation:

$$\mathrm{T}_{l1}^{l2}(\mathrm{t}, \theta_x, \theta_y, \theta_z) = \mathrm{TT(t)\ TX}(\theta_x)\ \mathrm{TY}(\theta_y)\ \mathrm{TZ}(\theta_z). \tag{3.22}$$

For the torso the rotation order is reversed to ensure that the heading orientation ($p^4 = \theta_z$) is first applied:

$$\mathrm{T}_t^{WCS}(\mathrm{t}, \theta_x, \theta_y, \theta_z) = \mathrm{TT(t)\ TZ}(\theta_z)\ \mathrm{TY}(\theta_y)\ \mathrm{TX}(\theta_x). \tag{3.23}$$

| Body part | Parameter | Limits [°] | Values($r^x + 1$) |
|---|---|---|---|
| Torso | orientation | [0,359] | 360 |
| | spline angle | [-10, 90] | 11 |
| | tilt angle | [-20, 20] | 5 |
| Head | gaze direction | [-45, 45] | 7 |
| | head angle | [-10, 30] | 5 |
| Left & right upper arm | arm frontal | [-60, 180] | 17 |
| | arm side | [0, 150] | 7 |
| | arm twist | [0, 135] | 4 |
| Left & right lower arm | elbow | [0, 135] | 10 |
| Left & right upper leg | leg frontal | [-45, 45] | 13 |
| | leg side | [0, 30] | 4 |
| | twist | [-90, 45] | 4 |
| Left & right lower leg | knee | [0, 90] | 7 |

Table 3.10: Parametrisation of body parts. Each parameter nominal range is defined by the physical constraints of the part, scaled into the range of $[0; r^x]$.

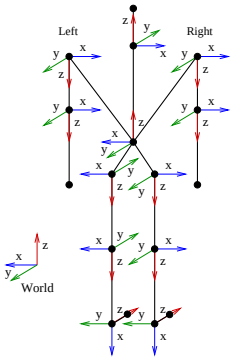



Figure 3.10: Neutral configuration of the LCSs.

The homogeneous rotation matrices with conventions of [214, pp.136–138] are:

$$\mathrm{TX}(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & \sin(\theta_x) & 0 \\ 0 & -\sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.24}$$

$$\mathrm{TY}(\theta_y) = \begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \tag{3.25}$$

$$\mathrm{TZ}(\theta_z) = \begin{bmatrix} \cos(\theta_z) & \sin(\theta_z) & 0 & 0 \\ -\sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.26}$$

The homogeneous translation matrix is

$$\mathrm{TT(t)} = \begin{bmatrix} 0 & 0 & 0 & t_x \\ 0 & 0 & 0 & t_y \\ 0 & 0 & 0 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.27}$$

The transition vector $\mathrm{t} = [t_x\, t_y\, t_z]^T$ is the constant position of the articulating joint relative to the parent limb, in the LCS of the parent. t is constant and unique for a human subject.

Finally, for any part $l1$, the transformation relative to WCS is the chain of individual limb to parent relative transformations:

$$\mathrm{T}_{l_1}^{WCS} = \mathrm{T}_{t}^{WCS} \ \ldots\ \mathrm{T}_{l_2}^{l_3}\, \mathrm{T}_{l_1}^{l_2}. \tag{3.28}$$

### 3.2.4 Body part projections

For pose evaluation, the AHHM is projected on the image, by means of individual parts. The projection of a point X on the frustum is given in equation (3.12). The set of which 3D points of the frustum, the sampling points, are projected is defined next. This set has only the visible points of the frustum (*i.e.* the front face).

If $2m$ generators are sampled, each with $n$ points, and $\mathrm{X}_i^j$ is the $i$-th ($i = 0..n$) sampling point on the $j$-th ($j = 1..2m$) generator of the frustum then

$$\mathrm{X}_i^j = \left[ R_M(i) \sin\left(2\pi \frac{j-1}{2m-1}\right) \quad R_m(i) \cos\left(2\pi \frac{j-1}{2m-1}\right) \quad \frac{i}{2n} \quad 1 \right]^T \tag{3.29}$$

where $R_M(i)$ and $R_m(i)$ are the radius of the elliptical cross-sections of the frustum at the level $i$:

$$R_M(i) = \mathrm{R}_{M,1} + (\mathrm{R}_{M,2} - \mathrm{R}_{M,1}) \frac{i}{2n} \ \text{ and} \tag{3.30}$$

$$R_m(i) = \mathrm{R}_{m,1} + (\mathrm{R}_{m,2} - \mathrm{R}_{m,1}) \frac{i}{2n}. \tag{3.31}$$

with $R_{M,k}$ the major, respective $R_{m,k}$ the minor base ellipsis radius of the two bases $k = 1, 2$. So defined, the two base point sets are $X_0^j$ and $X_n^j$, $j = 1..2m$.

To identify the visible generators, each point $X_0^j$ is compared to the opposite point $X_0^{j+m}$ on the same base; whichever has smaller depth is visible. If $X_0^j$ is visible, then generator $j$, and otherwise the opposite $j+m$ generator is visible. The pairwise comparison results in a circular list of visibility/invisibility of the generators with a single visible to occluded and one occluded to visible transition. The changes identify the edge generators resulting in the $2 \times (n + 1)$ points of edge set ${}^eX$, while all visible generators enclosed by the changes provide the set ${}^sX$ of the $(m + 1) \times (n + 1)$ silhouette points.

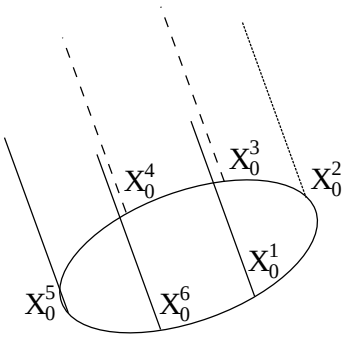



Figure 3.11: The projected visible edges are found first comparing pairwise the opposite points depth (*i.e.* $X_0^3$ with $X_0^6$), resulting the list of visibility *(visible, invisible, invisible, invisible, visible, visible)*, with transitions between $X_0^1$ and $X_0^2$ respectively $X_0^4$ and $X_0^5$. Since the segment $[X_0^2 \;\; X_0^5]$ is longer than $[X_0^1 \;\; X_0^4]$, the edge generators are $j \in \{2, 5\}$, and not the alternative $j \in \{1, 4\}$.

### 3.2.5 The self-occlusion reasoning

The depth map, figure 3.12, is an image the size of the input image, and with pixel values equal with the ID of the visible object (*i.e.* body part) at that pixel position. It is obtained by projecting one by one each body part $j$, in the order of increasing depth. If a pixel in the region of the body part is already marked then the body part at that location is occluded; otherwise is visible, and the pixel is marked accordingly $j$, the current body part. The computation of the depthmap is similar to Z-sorting, used in computer graphics.

To reduce the computations, the depth of a part is the depth of the frustum centre, and parts are projected onto the image as a quadrilaterals, defined by the edge generators $X^e$.

By the means of the depth map, all visible sampling points from the sets $X^e$ and $X^s$ are stored in the visible set, marked with hat, $\hat{X}^e$ respective $\hat{X}^s$.

Figure 3.12: Depth map example. Only the bounding rectangle of the person is shown.

Although is out the scope of the thesis, the visibility reasoning could be extended beyond self-occlusion to handle arbitrary objects. A prior scene model allows a depth to be built by projecting all scene objects and all the body parts in their decreasing depth order. The resulting map would contain the visible parts of objects and of the body. Occlusions by other persons or by dynamic objects could be solved similarly.

### 3.2.6 The Maximum Visibility prior

The Maximum Visibility prior prevents the projected model from being collapsed into a small region or, in the extreme case, into a point. For a limb it is

$$\pi_v^c(\mathrm{p}^\phi) = 1 - e^{-\frac{a^c(\mathrm{p}^\phi)}{n_a}}, \tag{3.32}$$

with $a^c(\mathrm{p}^\phi)$ the size of the rectangular area of the part projected onto the camera view $c$, surrounded by the two edge generators ${}^e\mathrm{X}^\phi$. It favours larger regions, while a zero area body parts have a zero prior probability. A normalising factor $n_a = 1000$ was used in our tests.

### 3.2.7 Three-dimensional humanoid structure test

To be valid, AHHM must allow a wide range of human poses. This is verified visually by generating all poses that result from advancing with unit steps over the valid range $[0; r^x]$ of a parameter $p^x$, with other parameters corresponding to the neutral pose ($p^i = 0$, for $i \neq x$). The generated poses have a wide diversity and are all valid. The subset of these, with the possible complete left leg configurations, is presented in figure 3.13.

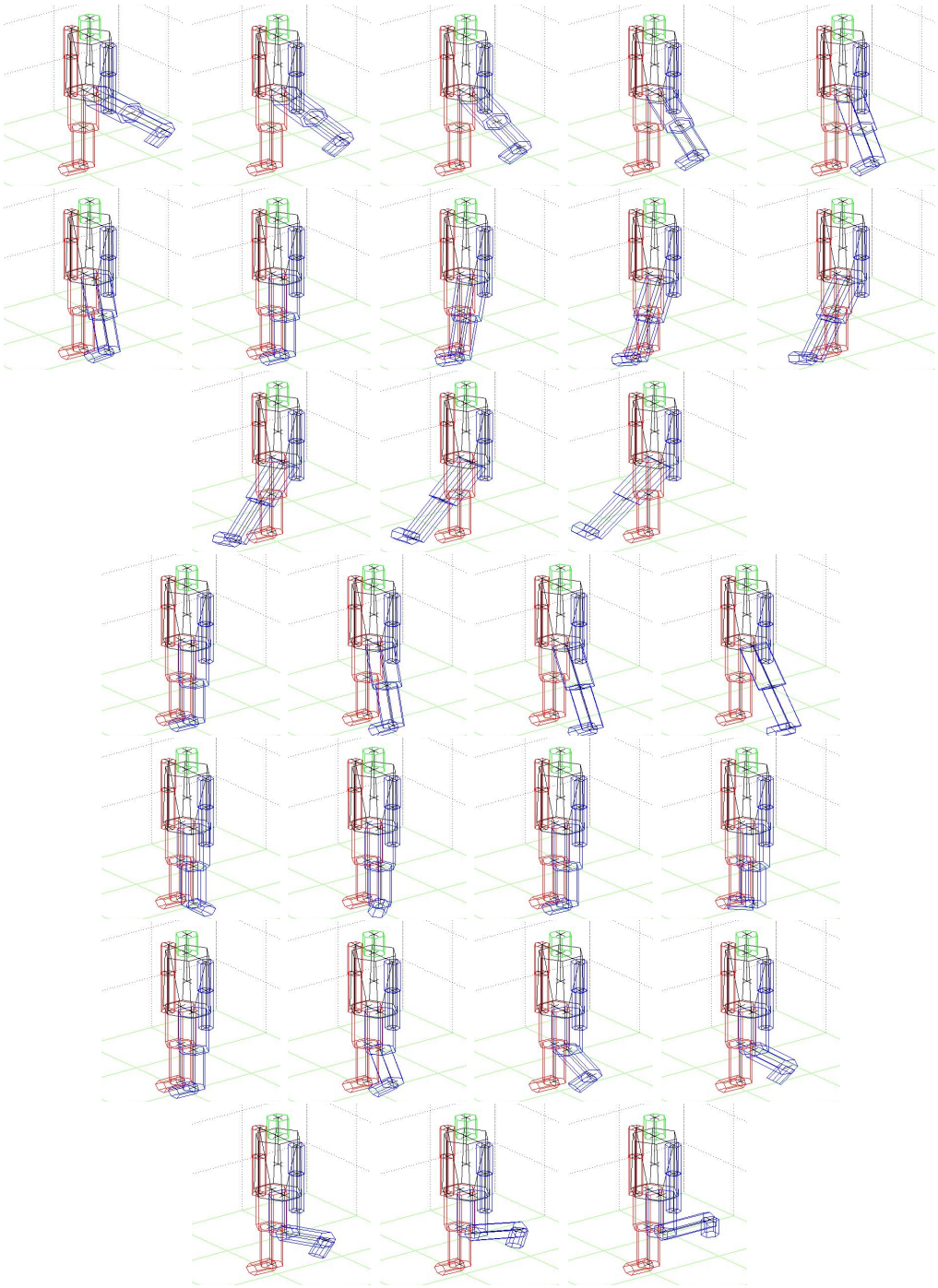



Figure 3.13: Left leg poses generated by individually altering the four parameters. These are assigned one by one each integer value in their range of $[0; r_x]$.

### 3.2.8 Comparison with HumanEva model

The HumanEva dataset (section 2.6.1) defines a purpose-built model, matching the motion capture system used for acquisition. This model was not given in the dataset documentation, however by reverse engineering it was found that it differs in several aspects from the AHHM (table 3.11). Both use an articulated structure and the same joints, however the

| Feature | AHHM | HumanEva |
|---|---|---|
| Number of body parts | 12 | 10 |
| Body part shape | elliptical frustums | cylinders |
| DOF per limb | 0–3 joint angles | 3 rotations & a translation |
| Coordinate systems | for each limb locally | global rotations |
| Limb length | constant | variable |
| Limb position relative to | parent-limb | root |

Table 3.11: Similarities and differences of the AHHM and HumanEva models.

AHHM has two additional body parts, two feet, that allow better recovery of the lower leg twist angle. The elliptical cross-sectioned frustums are expected to be more general models of body parts than cylinders. The body parts of AHHM have up to three *Degrees of Freedom* (DOF), while in HumanEva a limb transformations is an unrestricted rotation and translation, a $4 \times 4$ matrix, with a DOF up to 6.

Since the HumanEva dataset provides good training and evaluation sequences, it is the main training and testing data in chapter 4 to chapter 6. To be used, HumanEva parametrisation has to be converted to the AHHM. However, this is not straightforward. First, different LCS axis definitions are solved by rotating the HumanEva coordinates with the orthogonal angles from table 3.12. Each HumanEva global transformation $\mathrm{T}_l^{HE}$, to align the AHHM axis, is rotated with equation (3.33), where the two x and z rotations are from table 3.12.

$$\mathrm{T}_l^{WCS} = \mathrm{T}_l^{HE}\ \mathrm{TX}(\theta_x)\ \mathrm{TZ}(\theta_z). \tag{3.33}$$

From the global LCS to WCS transformations, the local LCS of limb $l1$ transformation to LCS of $l2$ is

$$\mathrm{T}_{l1}^{l2} = \left(\mathrm{T}_{l2}^{WCS}\right)^{-1}\ \mathrm{T}_{l1}^{WCS}. \tag{3.34}$$

Expanding equation (3.34) with the equations (3.24) to (3.27) and matching terms results the recovered angles

$$\theta_y = \arcsin(-\mathrm{T}_{1,3}), \tag{3.35}$$

| Part | $\theta_x$ | $\theta_z$ |
|---|---|---|
| Root | 0° | -90° |
| Head | 0° | 180° |
| Left upper arm | 180° | 180° |
| Left lower arm | 180° | 180° |
| Left upper leg | 0° | -90° |
| Left lower leg | 0° | -90° |
| Right upper arm | 180° | 0° |
| Right lower arm | 180° | 0° |
| Right upper leg | 0° | 90° |
| Right lower leg | 0° | 90° |

Table 3.12: HumanEva to AHHM transformations by rotations

$$\theta_x = \arctan(\mathrm{T}_{1,2}/\mathrm{T}_{1,1}), \tag{3.36}$$

$$\theta_z = \arctan(\mathrm{T}_{2,3}/\mathrm{T}_{3,3}). \tag{3.37}$$

If $\cos(\theta_y)$ is very small, due to the Gimbal lock effect, only $\theta_x + \theta_z$ can be computed, but not each individually. Assuming $\theta_x = 0$ results

$$\theta_z = \arctan(-\mathrm{T}_{2,1}/\mathrm{T}_{2,2}). \tag{3.38}$$

.

For the torso transformation equation (3.23), with the same computations, the rotations angles are

$$\theta_y = \arcsin(\mathrm{T}_{3,1}), \tag{3.39}$$

$$\theta_x = \arctan(-\mathrm{T}_{3,2}/\mathrm{T}_{3,3}), \tag{3.40}$$

$$\theta_z = \arctan(-\mathrm{T}_{2,1}/\mathrm{T}_{1,1}). \tag{3.41}$$

For small $\cos(\theta_y)$ results $\theta_x = 0$ and

$$\theta_z = \arctan(-\mathrm{T}_{1,2}/\mathrm{T}_{2,2}). \tag{3.42}$$

Specially for the elbow joint, with only one DOF, $\theta_x = \theta_z = 0$, therefore

$$\theta_y = \arctan(\mathrm{T}_{3,1}/\mathrm{T}_{1,1}). \tag{3.43}$$

The recovered angles matched with the relations from table 3.8 result in parametrisa-

tion in AHHM representation of the pose that was initially in the HumanEva format.

The joint positions t in equation (3.21), in the parent limb $l2$ LCS, is the last column of the transformation matrix $\mathrm{T}_{l1}^{l2}$. Since HumanEva has a sequence of training poses, the expected joint positions over the whole sequence are used.

Unfortunately, the conversion has multiple sources of errors. Fist, as mentioned above, the joint positions are assumed fixed in AHHM, but in HumanEva can change. If Gimbal lock is present then angles cannot be uniquely restored from the rotation matrix. The recovered angles do not satisfy the constant rotations expected in table 3.8, and a few large discrepancies suggest that the DOFs of the AHHM are too low to model all HumanEva poses. One, related to the upper leg twist angle, is solved by combining the leg twist angle with the recovered, non zero, knee twist that is assumed null in the AHHM.

Other conversion errors result from the markers used in motion capture, positioned on the loose clothing, and not at the physical joint (*i.e.* inside the limb); the physical joints are not on the limb symmetry axis, as assumed by both models, resulting in physically impossible poses, such as the backward bent knee shown in figure 3.14.

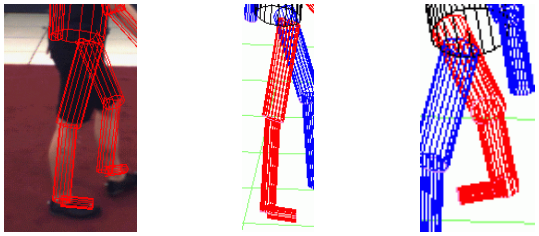


Figure 3.14: HumanEva dataset artefacts. Right leg bend is physically impossible in *S1 Walking 1* sequence, frame number 620.

The mean absolute error from equation (2.16) of the conversion from HumanEva into AHHM, computed over the whole dataset, results in a 3D error of 23 mm, equivalent to 3 pixels 2D mean error of the three camera views. Alternatively to the model conversion, tracking directly the HumanEva model [102,141,147] would result in lower tracking errors, however the AHHM development preceded the HumanEva dataset and therefore it was used.

## 3.3 Observation model

The observation model defines the likelihood $\lambda(\mathrm{O}|\mathrm{p})$ that is the conditional probability of observing O, given the pose p (section 2.4). Several methods for computing the likelihood

were reviewed in chapter 2, and was concluded that a compound likelihood is required. Therefore, an observation model consisting of multiple composition levels is used here.

The silhouette and edge based likelihoods are taken from Deutscher *et al.* [117, 139]. Deutscher's likelihoods were global, per pose, but here their definitions are body part-based, allowing for individual limb likelihoods to be computed and for self-occlusion to be considered.

### 3.3.1 Likelihood composition

The likelihood $\lambda(\mathrm{O}|\mathrm{p})$ of a pose is composed of multiple terms since

- multiple camera views provide the observation;
- multiple types of observation (*i.e.* silhouette, edge or colour) are derived;
- multiple body parts make up the pose.

Considering their independence, these sources are combined into the likelihood $\lambda(\mathrm{O}|\mathrm{p})$ as follows.

#### Camera views

Environmental conditions, camera characteristics, overlapping views, *etc.* suggest dependence of the camera views. However, in the same way as other authors [117, 137, 139, 141], the observation model considers the camera images independent. Therefore

$$\lambda(\mathrm{O}|\mathrm{p}) = \lambda(\{\mathrm{O}^j\}|\mathrm{p}) = \prod_{j=1}^{c} \lambda(\mathrm{O}^j|\mathrm{p}) \tag{3.44}$$

is the joint likelihood of the $c$ camera observations $\mathrm{O}^j, j \in \{1, \ldots, c\}$.

#### Body parts

As introduced in table 3.10, each body part is parametrised with a set of one to three parameters. The full PV or a parameter partition $\Phi$ may contain independent sub-partitions, defining body parts $\phi_k \subset \Phi$, with $\phi_k \cup \phi_l = \emptyset$. Considering the body part observations independent (comparably with [132, 136]), the joint likelihood of $\mathrm{p}^{\Phi}$ is the product of the

individual body part likelihoods

$$\lambda(\mathrm{O}^j|\mathrm{p}^\Phi) = \lambda(\mathrm{O}^j|\{\mathrm{p}^\phi\}_{\phi\in\Phi}) = \prod_{\phi\in\Phi} \lambda(\mathrm{O}^j|\mathrm{p}^\phi). \tag{3.45}$$

Through the hierarchical dependency in the AHHM and the self-occlusions, the part likelihoods are not independent. However, the first dependency is tackled by the hierarchical search of the particle filter (section 5.3); while the second dependency is solved with counting only visible parts while local likelihood is evaluated.

**Measurement type**

The observation $\mathrm{O}^j$ is based on the image provided by camera $j$. The acquired image is an RGB colour image, $\mathrm{I}_j$. It can generate the observation $\mathrm{O}^j = \{\mathrm{I}_1^j = f_1(\mathrm{I}^j), \mathrm{I}_2^j = f_2(\mathrm{I}^j), \ldots, \mathrm{I}_n^j = f_n(\mathrm{I}^j)\}$, a set of processed images, all resulting from $\mathrm{I}_j$. $f_i(\cdot)$ are known image processing functions for edge, silhouette extraction, colour enhancement, *etc.* On each, an independent likelihood can be defined, which together result in

$$\begin{aligned}
\lambda(\mathrm{O}|\mathrm{p}^\phi) &= \lambda(\mathrm{I}^j, f_1(\mathrm{I}^j), \ldots, f_n(\mathrm{I}^j)|\mathrm{p}^\phi) && (3.46)\\
&= \lambda(\mathrm{I}^j, \mathrm{I}_1^j, \ldots, \mathrm{I}_n^j|\mathrm{p}^\phi) && (3.47)\\
&= \prod_{i=1}^{n} \lambda_i(\mathrm{I}^j, \mathrm{I}_1^j, \ldots, \mathrm{I}_n^j|\mathrm{p}^\phi) && (3.48)\\
&= \prod_{i=1}^{n} \lambda_i(\mathrm{I}_i^j|\mathrm{p}^\phi). && (3.49)\\
& && (3.50)
\end{aligned}$$

The independence is again questionable, however it is assumed in the literature [185].

If edge, silhouette and colour images are used, $\mathrm{E}^j = \mathrm{I}_1^j$, $\mathrm{S}^j = \mathrm{I}_2^j$, $\mathrm{I}^j = \mathrm{I}_3^j$ equation (3.50) becomes

$$\lambda(\mathrm{O}^j|\mathrm{p}^\phi) = \lambda_s(\mathrm{S}^j|\mathrm{p}^\phi)\ \lambda_e(\mathrm{E}^j|\mathrm{p}^\phi)\ \lambda_c(\mathrm{I}^j|\mathrm{p}^\phi). \tag{3.51}$$

A well designed likelihood $\lambda_x$ is high for a good fit, if the projected $\mathrm{p}^\phi$ matches the image or if the body part is occluded and therefore the likelihood allows the fit, and leaves other camera likelihoods with better visibility to decide the hypothesis.

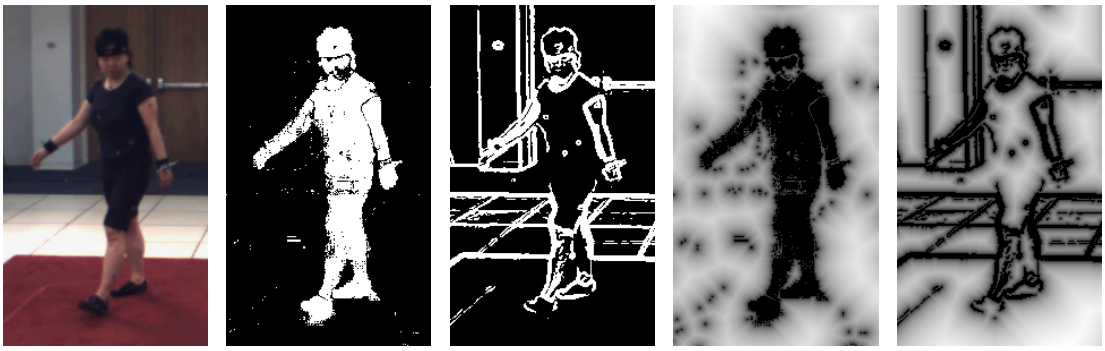
Finally, equations (3.44), (3.45( and (3.51) gives the expression of the local likelihood of $c$ views for the pose partition $\Phi$:

$$\lambda_l^{\Phi}(\mathrm{O}|\mathrm{p}) = \prod_{j=1}^{c} \prod_{\phi \in \Phi} \lambda_s(\mathrm{S}^j|\mathrm{p}^{\phi}) \lambda_e(\mathrm{E}^j|\mathrm{p}^{\phi}) \lambda_c(\mathrm{I}^j|\mathrm{p}^{\phi}). \tag{3.52}$$

Next, each component is defined.

### 3.3.2 Silhouette based likelihoods

To generate foreground image $\mathrm{S} = f_s(\mathrm{I})$, (figure 3.15(b)) the Stauffer and Grimson [177] background subtraction was used (see section 2.3.3). The Matlab implementation was applied with $K = 5$ Gaussians and $T = 0.1$ background threshold, pre-preprocessed offline the input sequences to speed up the tracking.



(a) (b) (c) (d) (e)

Figure 3.15: Multiple measurements. Initial colour image (a), silhouette image(b), edge image (c), silhouette based Chamfer distance (d) and edge based Chamfer distance(f).

The Chamfer distance transform, figure 3.15 (d) is a fast and robust method for computing distances from image features, therefore, in common with other authors [124,141] it is used for both silhouette and edge likelihood computations. The Chamfer distance chamf is computed with the forward-backward algorithm [215] during offline preprocessing.

The used expression for likelihood of the silhouette image S is

$$\lambda_s(\mathrm{S}|\mathrm{p}^{\phi}) = e^{-\frac{d_s(\mathrm{p}^{\phi})}{n_s}}, \tag{3.53}$$

where the distance

$$d_s(\mathrm{p}^{\phi}) = \underset{\tau \in {}^s\hat{\mathrm{X}}^{\phi}}{\mathcal{E}} [\mathrm{chamf}_{\mathrm{S}}(\tau)] \tag{3.54}$$

is the expectation of the Chamfer distance $\text{chamf}_\text{S}$, computed for the visible silhouette sampling points ${}^s\hat{\text{X}}^\phi$ of the body part $\phi$. The constant $n_s$ normalises the distance and $n_s = 5$ allows a standard deviation of the mean distances from the silhouettes equal to 2.23 pixels.

This silhouette likelihood is based on the Chamfer distance and not on the direct match of the binary silhouettes such as methods from table 2.9 do. The used expression was adapted from the edge likelihood, discussed in the next section.

The visibility of a sampling point is determined by the depth map (section 3.2.5). If all sampling points are on the silhouette or none of the sampling points are visible then the distance $d_s(\text{p}^\phi) = 0$, thus $\lambda_s(\text{S}|\text{p}^\phi) = 1$. When visible sampling points are outside the silhouette then the distance $d_s(\text{p})$ is high, resulting in a low $\lambda_s(\text{S}|\text{p}^\phi)$ likelihood.

### 3.3.3 Edge based likelihood

Edges images can be generated by multiple operators [210, pp.69–82]. Here, the edge image $\text{E} = f_e(\text{I})$, figure 3.15(c), is generated by the Sobel operator of Matlab `edge` algorithm, with 0.02 threshold and no thinning. The Sobel operator was chosen, since it allows quick edge extraction and it is ready implemented in many image processing libraries. The fit of body parts is generally not exact, therefore the used threshold and the not thinned edge responses allows higher uncertainty of the fit.

The likelihood of the edge image is similar to [26, 117, 141], *etc.*, the distance based methods from table 2.9, with the used expression

$$\lambda_e(\text{E}|\text{p}^\phi) = e^{-\frac{d_e(\text{p}^\phi)}{n_e}}, \tag{3.55}$$

where the distance

$$d_e(\text{p}^\phi) = \underset{\tau \in {}^e\hat{\text{X}}^\phi}{\mathcal{E}} [\text{chamf}_\text{E}(\tau)] \tag{3.56}$$

is the expectation of the Chamfer distance $\text{chamf}_\text{E}$ (figure 3.15(e)) computed for the visible edge sampling points ${}^e\hat{\text{X}}^\phi$ of the body part $\phi$. The constant $n_e$ normalises the distance, and $n_e = 40$ allows a standard deviations of the mean distances from the edges equal to 6.32 pixels. The $n_e$ is greater than $n_s$ because the edge image and hence the Chamfer distance is more noisy then the silhouette image.

### 3.3.4 Colour likelihood

Colour is a valuable clue for tracking [107, 110, 112, 133], although the measurements have problems with illumination changes, with multi-camera system colour inconsistencies, and with the major problem of initialisation. Therefore the colour likelihoods are computed for each camera independently, with different models, initialised in the first frame and continuously updated.

The used colour likelihood is similar to [110, 112] (see table 2.9 for colour histogram based likelihoods):

$$\lambda_c(\mathrm{I}|\mathrm{p}^\phi) = e^{-\frac{d_c(\mathrm{p}^\phi)}{nc}}, \tag{3.57}$$

with the Bhattacharyya distance

$$d_c(\mathrm{p}^\phi) = 1 - \sum_{y=1}^{bins} \sqrt{H_{\mathrm{p}^\phi}(y) H_M(y)} \tag{3.58}$$

of the current projection ($H_{\mathrm{p}^\phi}$) and the model ($H_M$) normalised colour histograms. Both are computed for the silhouette sampling points ${}^s\hat{\mathrm{X}}^\phi$ of the limb $\phi$:

$$H(y) = \frac{1}{nc} \sum_{\tau \in {}^s\hat{\mathrm{X}}^\phi} \delta[\mathtt{bin}(\tau) - y], \tag{3.59}$$

with the normalising constant

$$nc = \sum_{\tau \in {}^s\hat{\mathrm{X}}^\phi} 1. \tag{3.60}$$

The function $\mathtt{bin}(\tau)$ linearises the RGB colour space of a pixel $\tau \in \mathrm{B}$ into a bin number ($8 \times 8 \times 8$ bins were used). $\delta$ is the Kronecker delta function; $n_c$ normalises the distance with $n_c = 2$ providing an appropriate decay of the likelihood function.

The colour model $H_M$ is acquired for initialised areas of the body parts in the first image. To overcome the incorrect initialisation and the changing environmental illumination, the model is continuously updated. Updates are performed only if the body part $\phi$ silhouette and edge likelihoods show good confidence of the detection, being conditioned $\lambda_s(\mathrm{S}|\mathrm{p}^\phi) > \tau_s$ and $\lambda_e(\mathrm{E}|\mathrm{p}^\phi) > \tau_e$. It was found that $\tau_s = 0.7$ and $\tau_e = 0.9$ results in updating only when the edges and silhouettes fit well.

The updated new model $H_M$ is define by the running mean

$$H_M = (1 - \alpha_u)H_M + \alpha_u H_{\mathrm{p}^\phi}, \tag{3.61}$$

with the selected learning rate $\alpha_u = 0.6$.

### 3.3.5 Global likelihood

The global likelihood evaluates the pose considering all body parts. To avoid initialisation and update of the colour model, only silhouette and edge components are used. They are similar to local likelihoods (equations (3.53) and (3.55)), however the distance is the expectation of Chamfer distances over all body part sampling points $\Gamma = \bigcup_\phi {}^s\hat{\mathrm{X}}^\phi(\mathrm{p}))$ of the pose p:

$$\lambda_{Ge}(\mathrm{E}|\mathrm{p}) = e^{-\frac{d_{Ge}(\mathrm{p})}{n_e}} \text{ and} \tag{3.62}$$

$$\lambda_{Gs}(\mathrm{S}|\mathrm{p}) = e^{-\frac{d_{Gs}(\mathrm{p})}{n_s}}, \tag{3.63}$$

with $n_s = 4$ and $n_e = 0.5$ normalisation factor over the expected edge and silhouette distances:

$$d_{Ge}(\mathrm{p}) = \underset{\Gamma}{\mathcal{E}}[\bigcup_{\tau\in\Gamma} \mathrm{chamf}_{\mathrm{E}}(\tau)] \text{ respectively} \tag{3.64}$$

$$d_{Gs}(\mathrm{p}) = \underset{\Gamma}{\mathcal{E}}[\bigcup_{\tau\in\Gamma} \mathrm{chamf}_{\mathrm{S}}(\tau)] \tag{3.65}$$

.

Similarly to the local likelihoods, the multiple camera measurement of the combined edge and silhouettes, with equations (3.44) and (3.50) results the global likelihood:

$$\lambda_G(\mathrm{O}|\mathrm{p}) = \prod_{j=1}^{c} \lambda_{Ge}(\mathrm{E}^j|\mathrm{p})\, \lambda_{Gs}(\mathrm{S}^j|\mathrm{p}). \tag{3.66}$$

## 3.4 Summary

This chapter defines the necessary static prior models used in the rest of the thesis, as well as the observation models, consisting of complex combined likelihoods. The analysis of two calibration methods showed that calibration with projective geometry and vanishing

lines is inaccurate, therefore 3D to 2D point correspondences were used to manually post calibrate images without an acquired initial calibration.

Then, the structure and parametrisation of an articulated hierarchical human model was defined. This has simpler components and lower parameter space compared to other models. The projected pose formation and occlusion reasoning was presented, the model was compared to the HumanEva model, and the required conversion relations were recovered. The *hierarchical* structure of the AHHM will be exploited by the PF tracker in chapter 5.

Finally, the chapter defined the likelihoods of several image measurements conditioned by the pose. The likelihoods are local for limbs, based on edge, silhouette and colour, and are combined into multiple part or full body likelihoods in multiple views for the first time in the literature that we are aware of.

The camera model and the AHHM provides static priors for human tracking, while likelihoods are directly used by the tracker to evaluate hypotheses (chapter 5). The dynamic model that also includes the behavioural modelling is discussed next.

# Chapter 4

# Human dynamics and behaviour modelling

Because of its importance for security and surveillance, behaviour understanding is an ultimate goal of human tracking. Although research frequently focuses only on the sub-problem of tracking, this thesis addresses both tracking and behaviour understanding. In this chapter, methods are defined that will be applied both to human motion prediction in tracking and to the closely related behavioural analysis.

Behaviour analysis can be achieved without tracking and pose recovery [45,46], however arguments for analysing an intermediate, tracked human model were invoked in chapter 2. Here, important poses and movements are discovered and learnt to provide on the one hand the human dynamic (*i.e.* the motion) model, and on the other hand, a human-understandable description of the behaviour.

In contrast to other work [46, 52] behaviours are not restricted either to be periodic or to be global activities. They include both whole body activities (*e.g.* running, bending, standing, etc.) and local, detailed movements (*e.g.* right arm forward, left arm reaching).

The first part of the chapter proposes three dynamic models that are trained unsupervised and are able to generate random motion, not restricted to a specific action, but switching between activities. Various other dynamic models have been proposed, as shown in section 2.3.2. Like the models in this chapter, [166, 168, 170–172] tackle the extraction of poses and motions relevant to a human observer, but mainly for video segmentation and computer animation. Further, the importance of poses was also examined by [216],

however for a high dimensional mesh-body model.

The second part of the chapter builds on the discovered similar movements, and statistically learns their symbolic description for both detailed and global actions. In the perspective of integration with tracking, various aspects of the recognition rates are analysed.

## 4.1 Body feature vector and movement clusters

Pose, movement, action, activity, behaviour and gesture were defined in section 2.1.4. According to these definitions, poses and movements have no intentional content, while actions and activities have. Like an action or activity, behaviour has intentional content, but this is defined with respect to public approval or disapproval. A behaviour is therefore classified as accepted/allowed or denied/refused. Gestures are similar to actions, have an emotional content, and will not be analysed further in this thesis. As a result, behaviour and gestures are redundant, and the rest of this thesis focuses on pose, movement, action and activities. However, as defined in section 2.1.4, behavioural analysis refers to the whole process of symbolic description of the tracking data.

**Definition 9.** A *Body Feature Vector* (BFV) is a set of parameters describing the pose, with elements that are either *direct* pose parameters or features *derived* from the pose vector (PV).

In chapter 3 the *Articulated Hierarchical Human Model* (AHHM) is parametrised by the pose *Parameter Vector* (PV), p, defined in section 3.2.1. In this context, a partition $\phi$ of the PV,

$$\text{bfv} = \text{p}^{\phi} \tag{4.1}$$

is a BFV with direct parameters only (*i.e.* joint angle, body position, orientation). While the PV describes a whole pose, the BFV represents either a full or a partial pose (*e.g.* limbs), or by means of the derived parameters, other features such as length, velocity, acceleration or position vector. Therefore, for the general case, a multidimensional function $\mathtt{F} : \mathbb{R}^{24} \rightarrow \mathbb{R}^{k}$, possibly with memory, defines a BFV with derived features by operations over the pose parameters p:

$$\text{bfv} = \mathtt{F}(\text{p}). \tag{4.2}$$

As defined in section 2.1.4, the movement is a sequence of poses. Since a BFV describes

a full or partial pose, the movement

$$\mathrm{m} = [\mathrm{bfv}_{l_m-1}, \ldots, \mathrm{bfv}_1, \mathrm{bfv}_0] \tag{4.3}$$

is a sequence of consecutive BFVs with the current $\mathrm{bfv}_0$, and previous $\mathrm{bfv}_1, \ldots, \mathrm{bfv}_{l_m-1}$ poses. $l_m$ is the length or duration of the movement. Movements are the building elements of actions therefore movements can be parts of multiple actions.

**Definition 10.** *A Movement Cluster* (MC) is a set of similar movements.

MCs are discovered automatically, provided that training movements are sufficiently similar. Movements have no explicit representation in the real world, however the statistical probability of movements of a MC being an *Action A*, if learnt, gives the probability of an arbitrary movement from the cluster being part of the *Action A*.

To make an analogy with the English language, activities are sentences, actions are words, movements are letters of actions, and BFVs are the sequence of curves forming the letters. This structure is shown in figures 4.1 and 4.2.

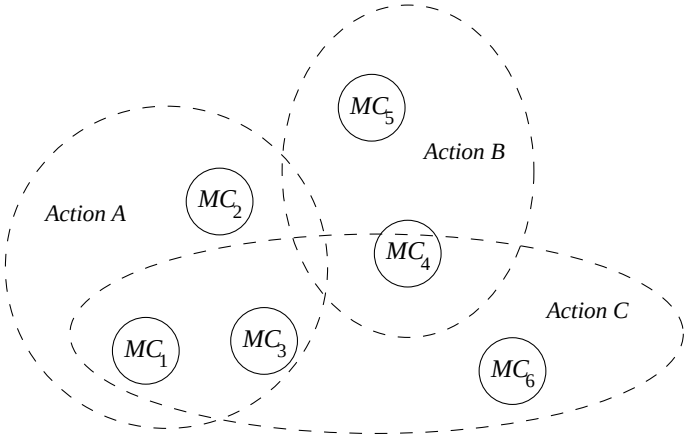



Figure 4.1: Movement clusters and actions. *Action A* results from any of the clusters $MC_1..MC_3$, *Action B* from $MC_4$ or $MC_3$, while *Action C* from $MC_1, MC_3, MC_4$ or $MC_6$. On the other hand a movement classified as $MC_1$ produces either *Action A* or *Action C*, *etc.*, with a probability characteristic to $MC_1$.

To exemplify the idea of MCs, figure 4.3 shows the distribution of the movements from the five activities of the HumanEva dataset and using a model with $n_{\mathcal{C}} = 60$ clusters. Although most of the movements are from the same activity, some MCs contain movements from more than one (*e.g.* MC number 15 contains movements from four *Jog*, *Box*, *Gesture* and *Throw/Catch* sequences, or MC number 39 contains movements from *Gesture* and

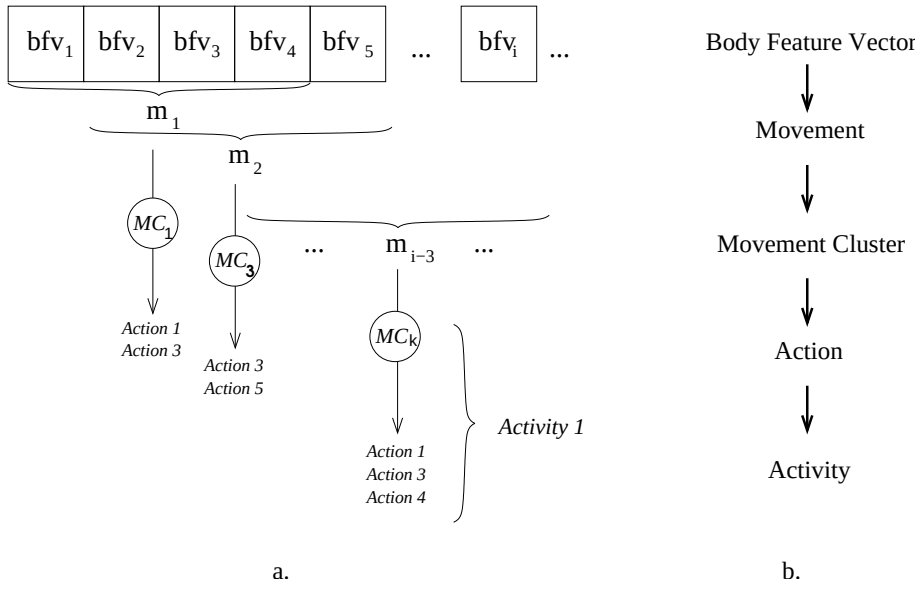



Figure 4.2: BFVs, movements, actions and activities. For an action primitive of length $l_m = 4$ body features $\text{bfv}_1, \ldots, \text{bfv}_4$ result in the movement $\text{m}_1$. The cluster $MC_1$, to which $\text{m}_1$ belongs, defines the possible actions (*i.e.* 1 and 3). Similarly, $\text{bfv}_2, \ldots, \text{bfv}_5$ define a different set of actions, *Action 3* and *5*, by means of $MC_3$. Presence or absence of actions (over a time), or the coexistence of different actions in a temporally ordered manner, result in activities.

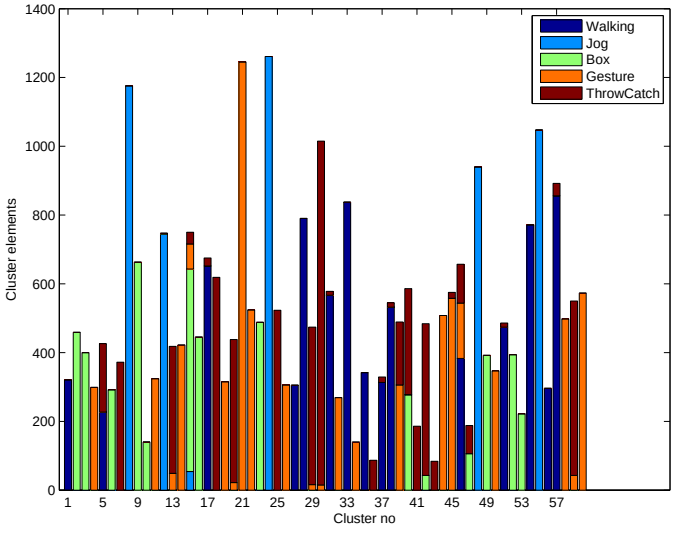



Figure 4.3: Distribution of actions for $n_{\mathcal{C}} = 60$ MCs. Some of the MCs contain only single type of actions (*e.g.* no. 1, 2, 3, 4, *etc.*) while others (*e.g.* no. 5, 13, 15, 20, *etc.*) contain movements from different sequences.

*Throw/Catch*). How MCs are generated and used for pose generation and behavioural analysis will be discussed later in section 4.5.

The approach taken in this thesis, to abstract into poses, movements, actions and activities, taken in this thesis, is similar to [24, 26, 31, 32], seen in section 2.1.4. All build semantic knowledge in a bottom up manner from simpler towards complex structures. The novelty in our approach is that the transition from the tracking data to symbolic description is performed through MCs, as will be explained further in this chapter.

Next, three dynamical models are introduced for modelling the human dynamics. One is extended, in section 4.6, for performing action recognition. The structure of the learning process is same for all three models, shown in figure 4.4. First, training data is compressed, then clusters are generated and finally their several probabilistic features are learnt. The blocks are explained in detail for the three models, followed by the visual verification if they are able to generate synthetic human motion.


Figure 4.4: Motion model learning overview. First, the MOCAP training data is compressed to reduce the number of the correlated body parameters. Then, based on similarities, clusters are formed from the alike training data. Finally, features of these cluster are learnt.

In the first two models, movements are one frame long, therefore they are equivalent to poses. The motion dynamics is represented by a pose to pose transition model.

## 4.2 Pose Transitional Model

Both poses and movements characterise an activity (section 2.1.1). Since movement is a sequence of poses, the *Pose Transition Model* (PTM) is motivated to capture the transitions between important poses and thus the motion.

The PTM is a *Hidden Markov Model* (HMM) with *Pose Cluster* (PC) states, similar poses each represented with a BFV. PTM assumes the Markovian property [217] that the probability of state at time $n+1$ is completely defined by the state at $n$, and therefore the probability of the next state state $pc_{n+1} = j$ is

$$\mathcal{P}\{pc_{n+1} = j \mid pc_n = i,\ pc_{n-1} = k \ldots pc_0 = m\} = \mathcal{P}\{pc_{n+1} = j \mid pc_n = i\} = \mathcal{T}_{i,j}. \quad (4.4)$$

One can argue that this assumption is weak for describing human motion. However, it will be shown that it is sufficient for simple predictions, and it is the basis of models with longer motion memory that are proposed in later sections. The transition probabilities define completely the transition from one PC at time $t$ to a new PC at $t+1$. These transitions could model movements or actions as more complex structures.

The learning process is shown in algorithm 1. The inputs are the BFVs, full or partial body poses of multiple training sequences of different activities, the number $n_{\mathcal{C}}$ of required clusters, and the compression factor, $\tau$. The output is the trained PTM model $\mathcal{M}$, with the clusters $\mathcal{M}.\mathcal{C}_c$ and the transitions $\mathcal{T}_{i,j}$ between these clusters The PTM is trained by learning the pose cluster $\mathcal{M}.\mathcal{C}_c$ (lines 1–5) and then the transitional probabilities $\mathcal{M}.\mathcal{T}_{i,j}$ (lines 6–10). The first involves data compression, clustering and statistical cluster modelling, while the second involves statistical modelling of the transitions. Each phase is explained below.

**Algorithm 1**: Pose Transition Model learning

**Input**: $\mathrm{S} = \bigcup_{i=sequences} \bigcup_{t=poses} \mathrm{bfv}_{i,t}$ – full pose database
$n_{\mathcal{C}}$ – number of clusters to generate
$\tau$ – minimum variance of selected eigenvectors
**Output**: $\mathcal{M}$ – motion model

1 $[\mathcal{M}.n_{pca} \quad \mathcal{M}.\mathrm{BT} \quad \mathcal{M}.BFV.\mu] = \texttt{PCA}(\{\mathrm{bfv}_{i,t}\}, \tau)$ `// compute PCA decomposition with energy` $\tau$
2 $[\{\mathcal{C}_{bfv_{i,t}}\}, \mathcal{M}.n_{\mathcal{C}}] = \texttt{Cluster}(\{\mathrm{bfv}_{i,t}\}, n_{\mathcal{C}})$ `// cluster with EM into` $n_{\mathcal{C}}$ `clusters`
3 **foreach** $\mathcal{C}_c$, $c = 1..\mathcal{M}.n_{\mathcal{C}}$ **do**
4 $\quad \mathcal{M}.\mathcal{C}_c.\mu = \mathcal{E} < \{\mathrm{bfv}_{i,t}\}_{\mathcal{C}_{bfv_{i,t}}=c} >$ `// mean of BFVs`
5 **end**
6 $\mathcal{T}_{i,j} = 0, 1 \leq i, j \leq \mathcal{M}.n_{\mathcal{C}}$
7 **foreach** $(\mathcal{C}_{bfv_{i,t}}, \mathcal{C}_{bfv_{i,t+1}}), \mathcal{C}_{bfv_{i,t}} \neq \mathcal{C}_{bfv_{i,t+1}}$ **do**
8 $\quad \mathcal{T}_{\mathcal{C}_{bfv_{i,t}}, \mathcal{C}_{bfv_{i,t+1}}} = \mathcal{T}_{\mathcal{C}_{bfv_{i,t}}, \mathcal{C}_{bfv_{i,t+1}}} + 1$
9 **end**
10 $\mathcal{M}.\mathcal{T}_{i,j} = \mathcal{T}_{i,j} / \sum_{i=1..\mathcal{M}.n_{\mathcal{C}}} (\mathcal{T}_{i,j})$ `// normalise emission probabilities`

The dynamic and behavioural models from this chapter are trained with the MOCAP data of the training partitions of the HumanEva dataset (see section 2.6.1). However, to match the lower frame rate videos, which are tracked and analysed in the next chapters, the training sequences are down-sampled from 60fps and 120fps to 20fps, comparable with 25fps CAVIAR and i-LIDS dataset frame rates. Therefore, each MOCAP sequence provides three respectively six training sequences.

### 4.2.1 Pose compression

In many activities, poses are inherently correlated with one another. While walking, for example, the right arm and left leg move forwards together as the left arm and right leg move backwards, and vice versa. In jogging, the correlations between left and right strides are similar, but the elbows have a sharper bending angle. *Principal Component Analysis* (PCA) is therefore applied to exploit these correlations and therefore to reduce the dimensionality of the model. At the same time, this reduces the evaluation and lookup times, and moderates time and memory requirements of the clustering algorithm for forming the PCs.

PCA reduces the dimensionality of the dataset vectors, by an orthogonally linear transformation and by removing the basis vectors (dimensions) with low contributions. This can be viewed as a coordinate system rotation, where the $d$-dimensional initial vector x is transformed into vector x$'$ in $k$-dimensional space. The manipulation is irreversible, since the new coordinate system basis loses $d - k$ dimensions. Transformation with the minimum reconstruction error [218] is achieved by PCA, and is given by the Karhunen-Loeve transform [184]:

$$\mathrm{x}' = \mathrm{BT}^t \cdot (\mathrm{x} - \mu), \tag{4.5}$$

where BT is the $d \times k$ matrix of eigenvectors, the basis mapping, and $\mu$ is the mean of the $d$ dimensional training data D.

To compute BT, this mean subtracted from the data

$$\mathrm{M} = \mathrm{D} - \mu, \tag{4.6}$$

and the eigenvalues($\lambda_i$) and eigenvectors ($\mathrm{e}_i$) of D are the solutions $\mathrm{x} = \mathrm{e}_i$ and $\lambda = \lambda_i$ of

$$(\mathrm{M} - \lambda \mathrm{I}) = \mathrm{x}. \tag{4.7}$$

The $d$ eigenvalues, $\lambda_i$, are provided by the characteristic equation

$$|\mathrm{M} - \lambda \mathrm{I}| = \lambda^d + a_1 \lambda^{d-1} + ... + a_{d-1}\lambda + a_d = 0, \tag{4.8}$$

while each eigenvector $\mathrm{e}_i$ is the root of the linear equation system

$$\mathrm{M}\mathrm{e}_i = \lambda_i \mathrm{e}_i. \tag{4.9}$$

If ordered by decreasing eigenvalues, the first $k$ eigenvectors are the columns of BT, and are the axes of the compressed coordinate system.

The total variance proportion of the first $k$ eigenvectors [219],

$$t_k = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}, \tag{4.10}$$

defines how accurately $k$ eigenvectors describe the $d$-dimensional space, providing a compression ratio of $\frac{k}{d}$.

This thesis uses the PCA code of the Netlab Matlab package [220], which calls the Matlab `eig` function for solving equation (4.8).

**Notation.** *The function*

$$[n_{pca} \quad \mathrm{BT} \quad \mu] = \texttt{PCAFunc}(\mathrm{D}, \tau) \tag{4.11}$$

*computes the most important $n_{pca}$ eigenvectors,* BT *with $t_{n_{pca}} \geq \tau > t_{n_{pca}-1}$, and the mean $\mu$ of the input data* D*, required in equation (4.5).*

**Notation.** *The* $\mathrm{pca}_\mathrm{x}$ *is the compact notation of the PCA projection with equation (4.5) of vector* x*:*

$$\mathrm{pca}_\mathrm{x} = \mathrm{BT}^t * (\mathrm{x} - \mu). \tag{4.12}$$

Table 4.1 shows the compression ratio and the variance proportion of the 18-dimensional BFV. The result is that seven PCA components represent with $t_7 = 91.41\%$ accuracy the 18 dimensional parameter space. The achieved compression ratio is 2.57, equivalent to a 61% data reduction of the orgininal data. For the tests in this thesis, an accuracy of $\tau = 95\%$ was chosen, which provides dimensionality reduction from 18 to 10, that is a 1.08 compression ratio (*i.e.* 44% reduction).

### 4.2.2 Cluster formation

The states of the PTM, the PCs are formed by clustering similar poses in the reduced PCA space. For clustering, *Expectation Maximisation* (EM) is used.

| PCA no. | Compr. ratio | Variance[%] | PCA no. | Compr. ratio | Variance[%] |
|---|---|---|---|---|---|
| 1 | 18.00 | 45.68 | 10 | 1.80 | 96.18 |
| 2 | 9.00 | 60.66 | 11 | 1.64 | 97.16 |
| 3 | 6.00 | 71.06 | 12 | 1.50 | 98.01 |
| 4 | 4.50 | 78.33 | 13 | 1.38 | 98.65 |
| 5 | 3.60 | 84.50 | 14 | 1.29 | 99.17 |
| 6 | 3.00 | 88.75 | 15 | 1.20 | 99.57 |
| 7 | 2.57 | 91.41 | 16 | 1.13 | 99.82 |
| 8 | 2.25 | 93.41 | 17 | 1.06 | 99.96 |
| 9 | 2.00 | 94.91 | 18 | 1.00 | 100.00 |

Table 4.1: PCA compression of poses. The compression ratio and the variance proportion of the 1 to 18 PCA components trained from 34451, 18-dimensional poses.

The general EM algorithm [184], algorithm 2, optimises the likelihood $\mathtt{Q}(\theta^x, \theta^y)$ of the new $\theta^x$ given the current $\theta^y$ estimate. It starts with an arbitrary $\theta^0$. First, in the **E-step**, $\mathtt{Q}(\theta, \theta^i)$ likelihoods are computed for all $\theta$ next estimates candidates, given the fixed $\theta^i$. Then, the **M-step** estimates the new $\theta^{i+1}$ that is $\theta$ with the highest likelihood.

**Algorithm 2**: Expectation-Maximisation

**Input**: $\theta^0$ – initial estimate;
$T$ – convergence criterion
**Output**: $\hat{\theta} = \theta^{i+1}$ – final estimate

$i = 0$
**repeat**
  $i = i + 1$;
  **E step:** compute $\mathtt{Q}(\theta, \theta^i)$;
  **M step:** $\theta^{i+1} = \operatorname{argmax} \mathtt{Q}(\theta, \theta^i)$;
**until** $\mathtt{Q}(\theta^{i+1}, \theta^i) - \mathtt{Q}(\theta, \theta^{i-1}) \leq T$ ;

Specifically for clustering, EM starts with a set of clusters, arbitrarily initialised, however frequently with *K-means clustering*. First, the membership function of each sample of all of the clusters is computed, then the new cluster centres are re-formed from the samples classified as members of the respective cluster.

In this thesis, the Matlab Expectation Maximisation Clustering algorithm of Frank Dellaert[1] was used. This applies *K-means* initialisation; assumes a Gaussian distribution for each cluster; and $\mathtt{Q}$ is the log likelihood defined with the Mahanabolis distance of the training data to current cluster centres.

From the set of input vectors $\{D_i\}$, the function $[\{\mathcal{C}_{D_i}\}, n^*_{\mathcal{C}}] = \mathtt{Cluster}(\{D_i\}, n_{\mathcal{C}})$ attempts to generate $n_{\mathcal{C}}$ clusters and returns the cluster number $\mathcal{C}_{D_i}$ of each $D_i$. If any of

[1] available from `http://www-static.cc.gatech.edu/~dellaert`

the cluster fails to have a number of members equal to the compressed data dimensionality (*i.e.* the covariance of the cluster is not defined), then clustering is repeated. After five failed attempts (with different stochastic initialisation of the K-means), cluster number is reduced by 2%, arbitrary choosen, and clustering restarts. The number of clusters found is $n^*_{\mathcal{C}}$, $n^*_{\mathcal{C}} \leq n_{\mathcal{C}}$.

Iterative Minimum Squared-Error Clustering and Hierarchical Clustering [184] are alternatives for generating the clusters, but considering the large training dataset the faster EM was chosen. Minimising multiple entropies of the data [166] is also an option, however the method is better suited to offline segmentation of a single, whole sequence. Methods such as approximate K-means or hierarchical K-means [221] could improve the clustering speed and give more compact clusters of similar poses or movements

### 4.2.3 Cluster modelling

Returning to PTM learning, algorithm 1, after each BFV is classified into one of the clusters $\mathcal{C}_{bfv_{i,t}}$, the mean BFV, $\mathcal{C}_c.\mu$ is computed in lines 3–5 from all BFVs forming cluster $\mathcal{C}_c$. This mean allows a simple representation (*e.g.* for visualisation purposes) of each PC.

### 4.2.4 Transition probabilities

Transition probabilities are the normalised PC transition frequencies $\mathcal{C}_{BFV_{i,t}} \rightarrow \mathcal{C}_{BFV_{i,t+1}}$, represented by the $\mathcal{T}$ matrix. The non-self transitions are computed in lines 6–10 of the learning algorithm 1. The self-transitions, $\mathcal{C}_{BFV_{i,t}} = \mathcal{C}_{BFV_{i,t+1}}$, are frequent, but do not add motion information, and so are removed, although this has the disadvantage that the length of the motion is lost. The temporal transition between poses will be modelled in section 4.3 with two different motion models.

### 4.2.5 Algorithm output

The PTM, generated by algorithm 1, is composed of pose clusters and PC transitions. The model $\mathcal{M}$ consists of:

- $n_{\mathcal{C}}$, number of clusters,
- $n_{pca}$, number of PCA components used for model compression,

- BT, projection vector onto the PCA space,
- $BFV.\mu$, BFV mean value for PCA conversion,
- $\mathcal{T}$, transition probabilities between PCs,
- and for each cluster $\mathcal{C}_c$, the mean $\mathcal{C}_c.\mu$ of the Gaussian distribution model.

### 4.2.6 Synthetic motion generation

If the model is good, then it creates pose sequences resembling human motion. This is tested by exploring the PC space with the learnt transition probability. Therefore, starting from an arbitrary pose cluster $pc_0$, with the learnt transition probabilities $\mathcal{T}$, $\{pc_1, pc_2, \ldots, pc_t\}$ are generated iteratively. Each PC sample is represented visually with the learnt mean BFV pose, $\mathcal{C}_{pc}.\mu$.

The motion from figure 4.5 is generated by a model trained for $n_{\mathcal{C}} = 100$ clusters. It is a sequence resembling *Walk* activity, with the first 20 poses of the 100-pose long synthetic sequence from figure 4.6 that visualises the transitions between the PC as a graph of numbered ellipses. The starting pose ($pc = 3$) defines the resulting initial motion as walking. However being a random process, change of activity is allowed at any instance. This happens in $pc = 6$ (figure 4.6) and the activity switches to *Box*, clear from the visual inspection of the later synthetic motion. Loops of the state transition emphasise the periodicity of the generated motion, and the transitions between loops show that the model switches from one action to the other in a way that resembles realistic action switches in human activities.

The randomness does not allow a smooth motion, and this disrupted behaviour is accentuated by the gaps between the limited numbers of discretized PCs poses. This model is restricted because the fixed and finite PCs poorly represent the pose state space. Replicating the model, using several BFVs, is one solution, and will be analysed later in this chapter in the context of temporal models. However, to model smooth transitions between the PCs, two models that account for the temporal relation of the poses are introduced next.

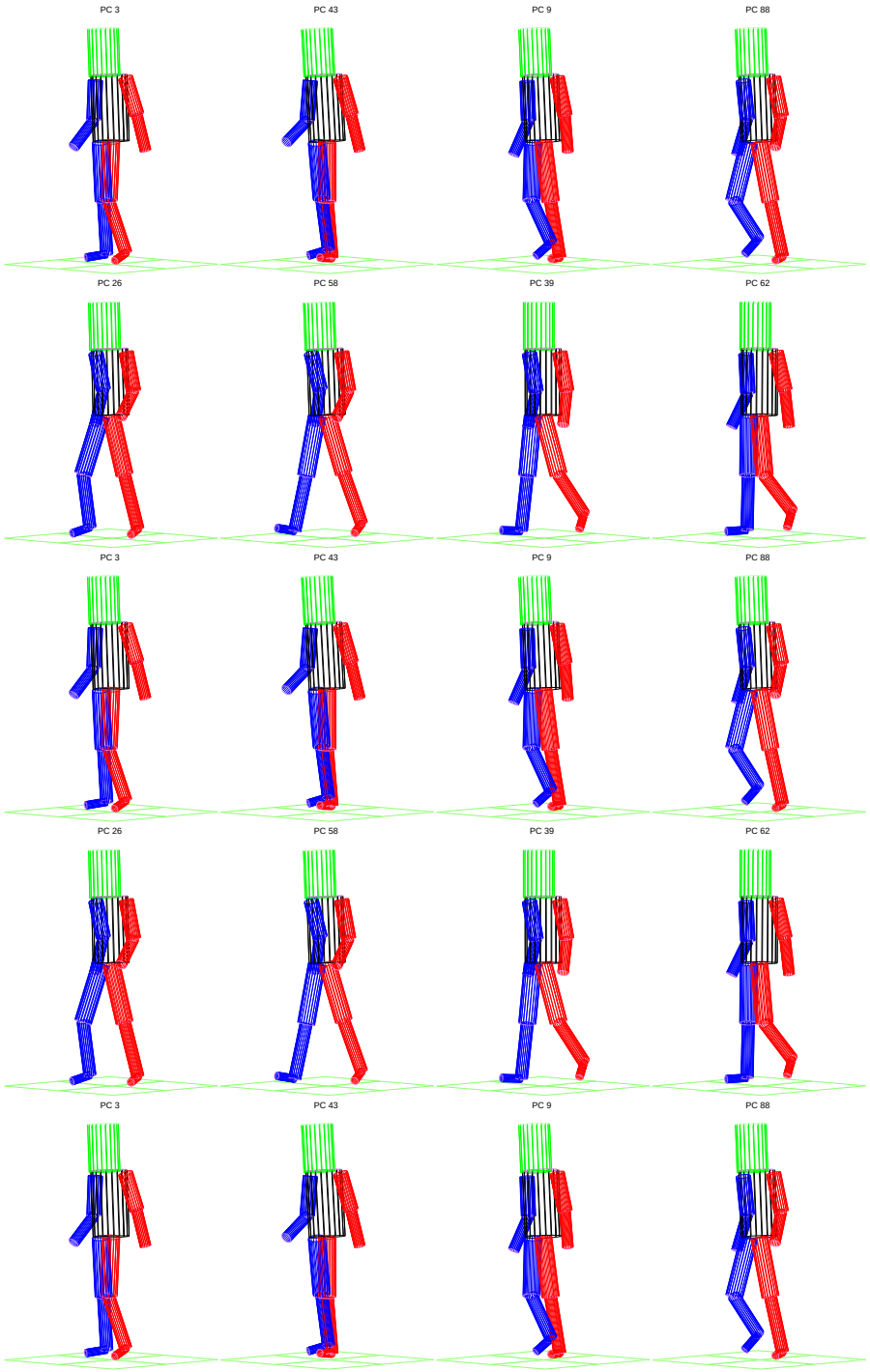



Figure 4.5: First 20 poses of random motion generated with the PTM ($n_{\mathcal{C}} = 100$) [◊].

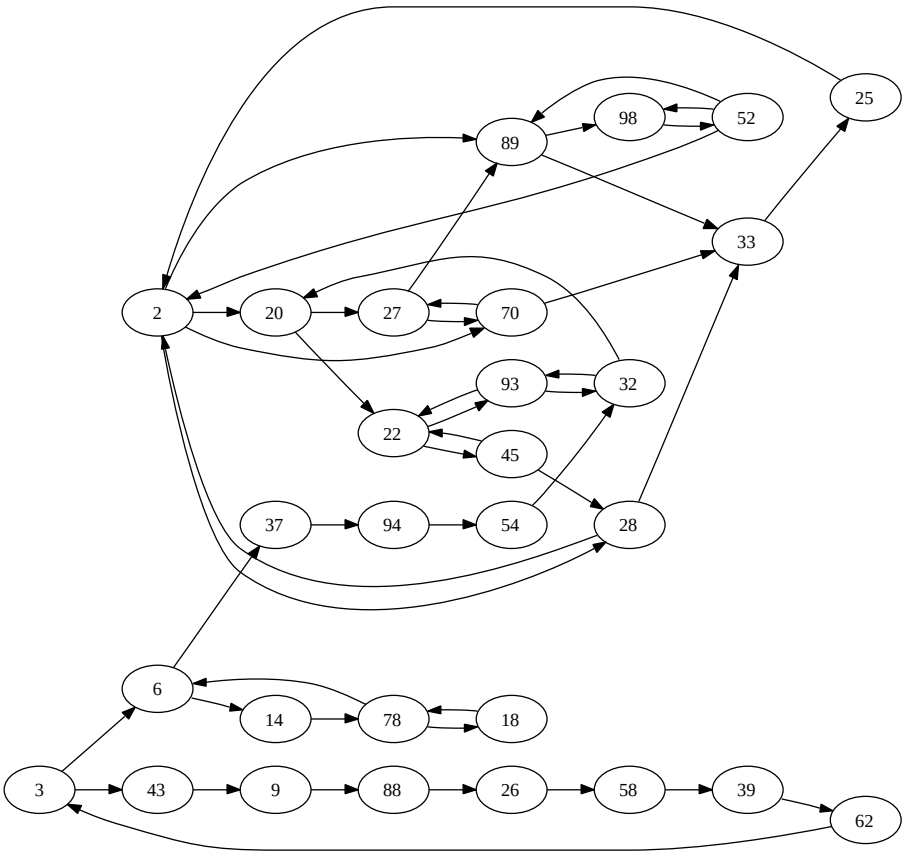



Figure 4.6: A transition sequence with PTM. The graph of 100 randomly generated transitions between PCs, with the first 20 poses shown in figure 4.5. The transitions include loops and pass trough a limited number of PCs.

## 4.3 Continuous transition models

HMMs have been used [26,32] to detect human acts and activities. However, like the PTM, these fail because the limited number of pose states and discrete transitions between them cannot capture the continuity of the motion, as previous section showed. Further, it is controversial whether a single pose alone can define the next, as the Markovian model requires. For example, a neutral leg-pose in the next instance can change into forward, backward, left and right pose. However, one additional previous pose defines the movement direction, and therefore from the four choices one is more likely. Other problems are the changing frame rate, *e.g.* camera acquisition frame rate is different for HumanEva and CAVIAR datasets, and motion speed. If the motion speeds up then transition probabilities should also increase or skip intermediate poses. The dimensionality of the human model parameter space is high. Hence, proper modelling requires a large number of discrete pose states to represent the continuum. To alleviate these, two alternative models were developed, both preserving the basic Markovian model providing a compatible and low complexity prediction for Particle Filter tracking (chapter 5).

## 4.4 Continuous Time Pose Transition Model

For the PTM, the transitions between two PC are instantaneous. In contrast, for the *Continuous Time Pose Transition Model* (CTPTM) the transitions have learnt lengths. When motion is generated with CTPTM, the transitions are performed with this duration.

Algorithm 3 shows the learning process. It has the same PC clustering of the PCA reduced pose database as PTM (lines 1–5), however for each PC transition $i$ to $j$, not only transition probabilities but also durations, $\mathcal{M}.Duration_{i,j}$, are learnt by their Gaussian mean, $\mu$, and variance, $\mathbf{P}$. Therefore, an arc between PCs represents a continuous sequence of poses, *i.e.* a movement, delimited by the starting and ending poses.

The transition duration, defined in figure 4.7, is the half length of the initial and the adjacent PCs. These durations are the number of consecutive BFVs classified into the $pc_j$, and respectively the next $pc_k$ clusters.

Figure 4.8 is an example of a model with 20 clusters learnt from the HumanEva training data. It shows the complexity of a such model, each PC having 3–4 inbound and the same outbound transitions.

**Algorithm 3**: CTPTM learning

**Input**: $\mathrm{S} = \bigcup_{i=sequences} \bigcup_{t=poses} \mathrm{bfv}_{i,t}$ – full pose database
$n_{\mathcal{C}}$ – number of clusters to generate
$\tau$ – minimum variance of selected eigenvectors

**Output**: $\mathcal{M}$ – motion model

$[\mathcal{M}.n_{pca} \quad \mathcal{M}.\mathrm{BT} \quad \mathcal{M}.BFV.\mu] =$ `PCA(`$\{\mathrm{bfv}_{i,t}\}$, $\tau$`)` `// compute PCA decomposition`

`// with energy` $\tau$

$[\{\mathcal{C}_{bfv_{i,t}}\}, \mathcal{M}.n_{\mathcal{C}}] =$ `Cluster(`$\{\mathrm{bfv}_{i,t}\}$, $n_{\mathcal{C}}$`)` `// cluster with EM into` $n_{\mathcal{C}}$ `clusters`

**foreach** $\mathcal{C}_c$, $c = 1..n_{\mathcal{C}}$ **do**

    $\mathcal{M}.\mathcal{C}_c.\mu = \mathcal{E} < \{\mathrm{bfv}_{i,t}\}_{\mathcal{C}_{BFV_{i,t}}=c} >$ `// mean of BFVs`

**end**

$\mathcal{M}.\mathcal{T}_{i,j} = 0$ for all $1 \leq i, j \leq n_{\mathcal{C}}$

$t_{i,j} = \emptyset$ `// initialise transition duration` $i \rightarrow j$

**foreach** $(\mathcal{C}_{bfv_{i,t}}, \mathcal{C}_{bfv_{i,t+1}}), \mathcal{C}_{bfv_t} \neq \mathcal{C}_{bfv_{i,t+1}}$ **do**

    $t_1 =$ length of states $p_t$, before (inclusive) $t$

    $t_2 =$ length of states $p^{t+1}$, after (inclusive) $t+1$

    $t_{\mathcal{C}_{bfv_{i,t}}, \mathcal{C}_{bfv_{i,t+1}}} = [t_{\mathcal{C}_{bfv_{i,t}}, \mathcal{C}_{bfv_{i,t+1}}}; \quad (t_1 + t_2)/2]$ `// add` $\mathcal{C}_{bfv_{i,t}} \rightarrow \mathcal{C}_{bfv_{i,t+1}}$ `// transition time to the list of times`

    $\mathcal{T}_{\mathcal{C}_{bfv_{i,t}}, \mathcal{C}_{bfv_{i,t+1}}} = \mathcal{T}_{\mathcal{C}_{bfv_{i,t+1}}, \mathcal{C}_{bfv_{i,t+1}}} + 1$

**end**

$\mathcal{M}.\mathcal{T}_{i,j} = \frac{\mathcal{T}_{i,j}}{\sum_{i=1..n_{\mathcal{C}}}(\mathcal{T}_{i,j})}$ `// normalise outgoing probabilities`

$\mathcal{M}.Duration_{i,j}.\mu = \mathcal{E} < t_{i,j} >)$ for all $1 \leq i, j \leq n_{\mathcal{C}}$

$\mathcal{M}.Duration_{i,j}.\mathbf{P} = \mathrm{var} < t_{i,j} >$ for all $1 \leq i, j \leq n_{\mathcal{C}}$

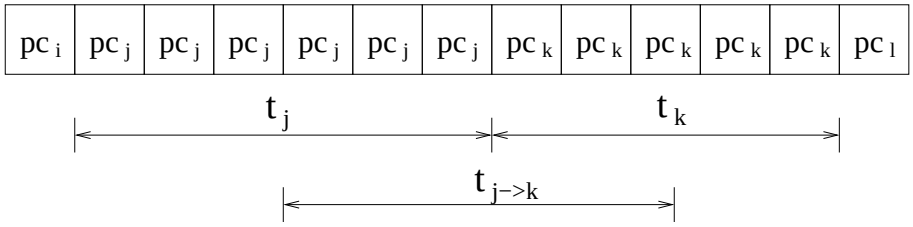



Figure 4.7: Transition duration definition. Transition from $PC_j$ to $PC_k$ is the mean duration $t_{j \rightarrow k} = 0.5(t_j + t_k)$.

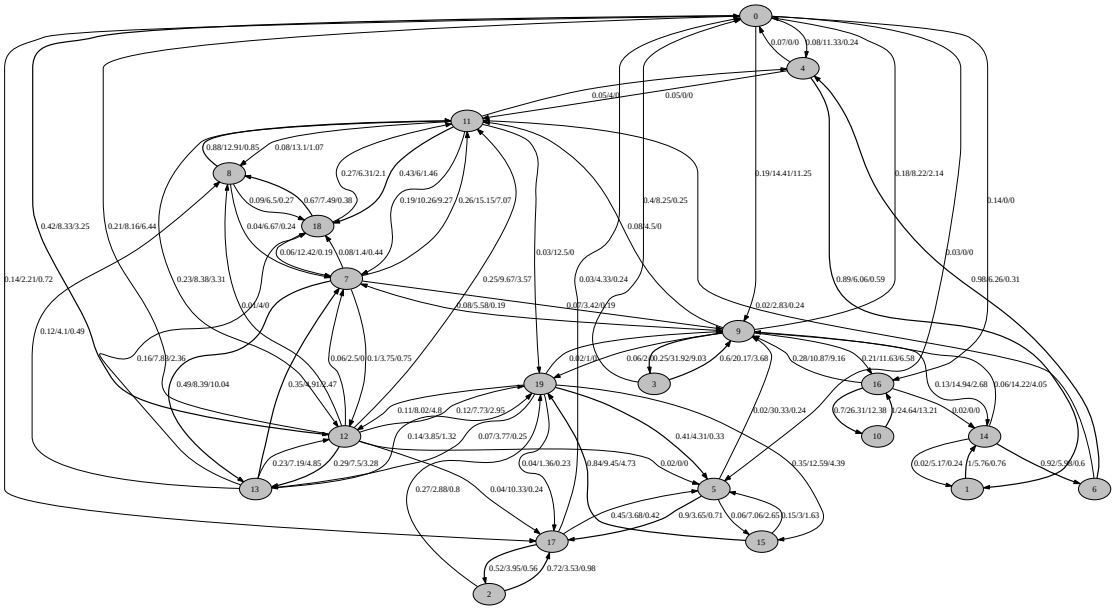



Figure 4.8: Continuous Time Pose Transition Model. Edge labels show as $\alpha/\beta/\gamma$ the transition probabilities ($\alpha$), the mean ($\beta$) and variance ($\gamma$) of the transition for a model with $n_{\mathcal{C}} = 20$ clusters.

### 4.4.1 Synthetic motion generation

With the CTPTM, synthetic motion generation is provided by algorithm 4. It is similar to PTM generation, except that the simulation maintains not only the current state, but also the next state $pose_{next}$ and the time $c$ spent in the current PC. When this duration reaches the transition time $t_{transition}$, the next state becomes the current, and a succeeding PC is generated with the probability $\mathcal{T}_{pose,pose_{next}}$ together with the new transition time $t_{transition}$. The transition time is sampled (in line 8) from the learnt distribution. The sampling is described later in section 5.2.1 together other stochastic algorithms.

The current pose $\text{bfv}_{t+1}$ is the linear mix of the current and next PC mean poses ($\mathcal{C}_c.\mu$) with a ratio of $c/t_{transition}$. The increment $dt$ is advantageous for motion speed alteration, especially if the training data has a different sampling rate from the generated motion.

Figure 4.9 illustrates the random transition sequence between the first 100 poses generated with a CTPTM with $n_{\mathcal{C}} = 100$ PCs. The diagram visualises the periodicity of the generated motion, composed from 15 PC. Further, figure 4.10 shows the generated first 20 poses resembling a normal walking motion.

The CTPTM generates smooth motion, and since transitions correspond to activities, it could be used for activity recognition or motion generation of a specific activity.

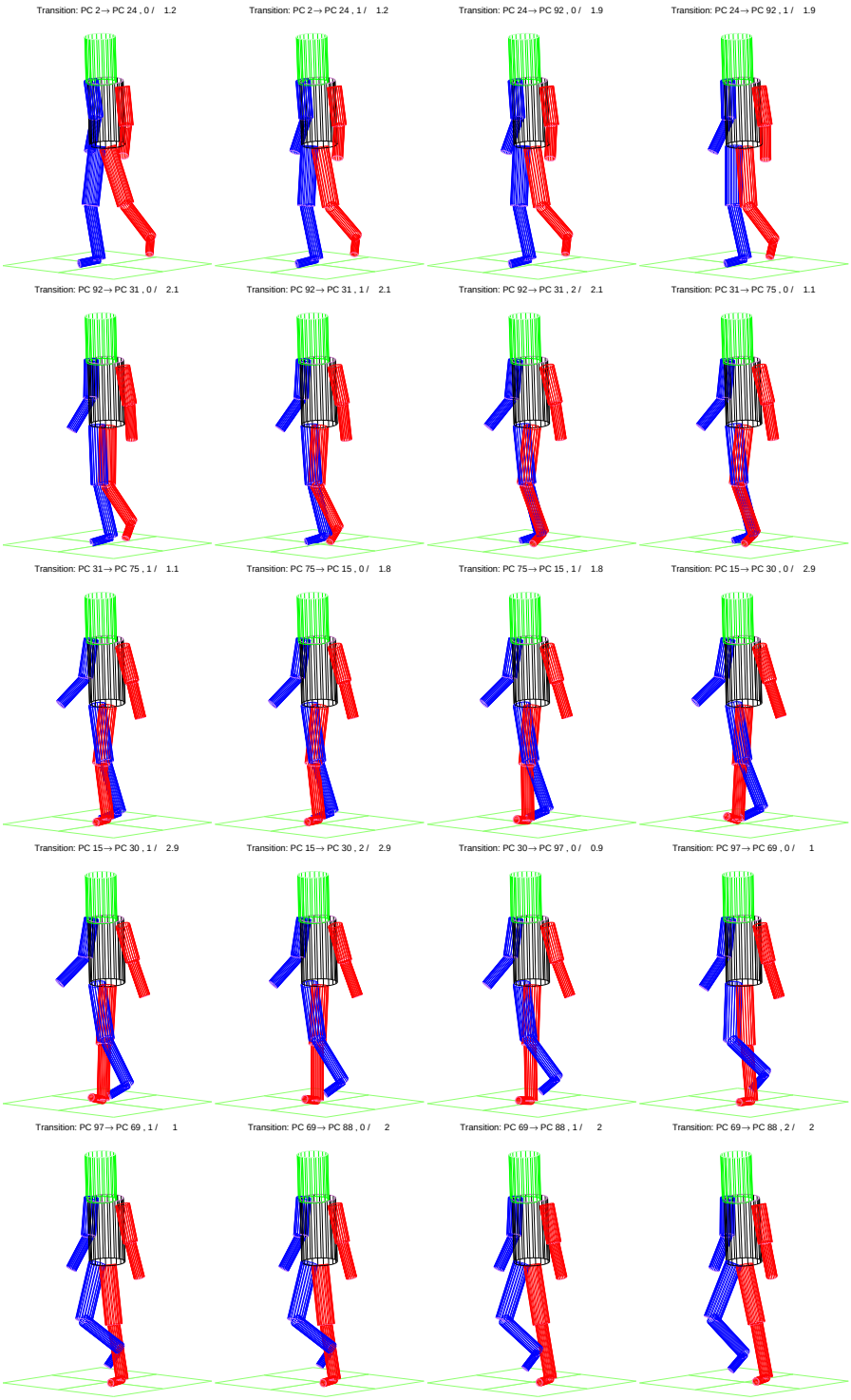



Figure 4.9: Random motion generated with the CTPTM ($n_{\mathcal{C}} = 100$). For the first 20 poses, the current and the next PC are shown, together with the current time / total transition time [◊].

**Algorithm 4**: CTPTM synthetic motion generator

**Input**: $\mathcal{M}$ – model
$dt$ – frame rate increment, with $dt = 1$ identical with the model frame rate
$pc$ – initial pose cluster
**Output**: $\{\text{bfv}_t\}$ – the set of generated BFV (poses)

$t = 0$
$c = 0$ `// transition time counter`
$t_{transition} = -1$ `// next transition time`
**repeat**
  **if** $t_{transition} \leq c$ **then**
    $pc = pc_{next}$
    $pc_{next}$= select one with probability $\mathcal{M}.\mathcal{T}_{pc,pc_{next}}$
    $t_{transition} \sim \mathcal{N}(t; \mathcal{M}.Duration_{pc,pc_{next}}.\mu, \mathcal{M}.Duration_{pc,pc_{next}}.\mathbf{P})$
    $c = 0$
  **end**
  $r = c/t_{transition}$
  $\text{bfv}_{t+1} = (1 - r) \cdot \mathcal{M}.\mathcal{C}_{pc}.\mu + r \cdot \mathcal{M}.\mathcal{C}_{pc_{next}}.\mu$
  $t = t + 1$
  $c = c + dt$
**until**

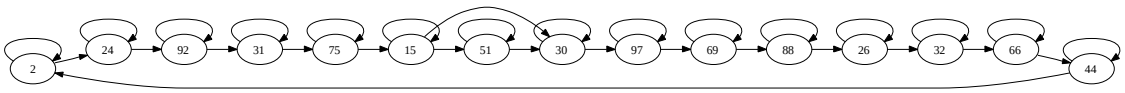



Figure 4.10: A transition sequence with CTPTM. The graph of 100 randomly generated transitions between PCs, with the first 20 poses shown in figure 4.9.

## 4.5 Movement Cluster Model

The *Movement Cluster Model* (MCM) extends the PTM by adding a pose history. For smoother transitions between poses, the pose clusters are replaced by clusters of similar movements (movement clusters, MC). For traditional HMMs, the number of states and the transitions are discrete and finite. However, in the MCM states and transitions are continuous and infinite, represented by Gaussian distributions of the current and the next state.

Considering $l_m + 1$ long movements, the clustered MCs represent the most similar movements. If each movement that is a member of a cluster is separated into the first $l_m$ and the last BFV then the $l_m$ long movements define the transition to the new BFV, the last item of the clustered movement. The $l_m$ long MCs are characterised with a Gaussian distribution (*i.e.* the mean and covariance) of the PCA compressed movements. A similar statistical representation of the *next BFV* (NBFV) is employed.

For a transition, the model first seeks for the current MC, and the MC with the minimum distance from the current compressed movement. The NBFV is then estimated that, concatenated with the active movement, results in the next movement, and the process is repeated. This mechanism defines the transition, illustrated in figure 4.11. Unlike a HMM, where transition probabilities from one state can be directly drawn, in the MCM the transitions are hidden by the NBFV Gaussian model. However, they provide a continuous parameter estimation and define the transition using the movement, not only a single pose.

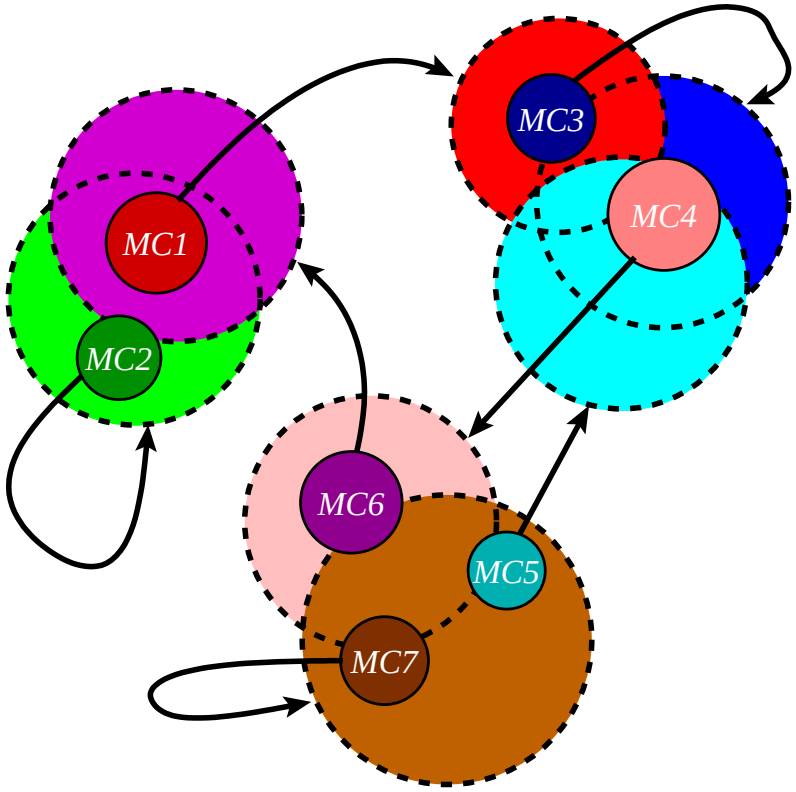



Figure 4.11: Visual example of a MCM. The MCs are represented by smaller continuous disks *MCi*. The MC generates a new BFV suggested by larger, dotted circle and with all but the first BFV of the the current MC results in the new movement that is classified into a MC. *MC1* transforms into *MC3* or *MC4*; *MC2* transforms into *MC1* or stays in the same state (but changes the parameter values); *MC3* to *MC3* or *MC4*; *MC4* to *MC5*, *MC6* or *MC7*; *MC5* to *MC4*;*MC6* to *MC1* or *MC2*; and *MC7* to *MC5*, *MC6* or *MC7*. Any transition will result in changes of the continuous BFV parameters.

### 4.5.1 Compression of movements

The opportunity for compressing BFVs was shown in section 4.2.1. The compression allows compact data representation and lowers the memory requirement for clustering. In

addition to the inter BFV parameter correlation, component BFVs from a movement are correlated, since a pose is strongly related to the preceding and following poses.

Table 4.2 shows the compression ratio and the variance proportion of the $10 \times 18$D pose sequences (*i.e.* movements). Seven PCA components represent the data with an accuracy of $t_7 = 90.44\%$ with a compression ratio of 25.71. For an accuracy of $\tau = 95\%$, the $t_{10} = 95.06 > \tau > t_9 = 93.82$ condition is satisfied by $k = 10$ eigenvalues, compressing to 6% of the original movement length, that is a compression ratio of 30.0. For the uncompressed $l_m \times 18$ dimensional BFV data of the S1 and S2 HumanEva subjects, the compression ratio varies from 2.25 ($l_m = 1$) to 42 ($l_m = 35$) for an accuracy of $\tau = 95\%$.

| **PCA no.** | **Compr. ratio** | **Variance[%]** | **PCA no** | **Compr. ratio** | **Variance[%]** |
|---|---|---|---|---|---|
| 1 | 180.00 | 45.51 | 25 | 7.20 | 99.73 |
| 2 | 90.00 | 60.32 | 27 | 6.67 | 99.80 |
| 3 | 60.00 | 70.53 | 29 | 6.21 | 99.85 |
| 4 | 45.00 | 77.60 | 31 | 5.81 | 99.89 |
| 5 | 36.00 | 83.64 | 33 | 5.45 | 99.91 |
| 6 | 30.00 | 87.82 | 35 | 5.14 | 99.92 |
| 7 | 25.71 | 90.44 | 37 | 4.86 | 99.94 |
| 8 | 22.50 | 92.38 | 39 | 4.62 | 99.95 |
| 9 | 20.00 | 93.82 | 41 | 4.39 | 99.96 |
| 10 | 18.00 | 95.08 | 43 | 4.19 | 99.97 |
| 11 | 16.36 | 96.03 | 45 | 4.00 | 99.97 |
| 12 | 15.00 | 96.85 | 46 | 3.91 | 99.97 |
| 13 | 13.85 | 97.48 | 47 | 3.83 | 99.98 |
| 14 | 12.86 | 97.98 | .. | .. | ..... |
| 15 | 12.00 | 98.37 | 51 | 3.53 | 99.98 |
| 17 | 10.59 | 98.99 | 53 | 3.40 | 99.99 |
| 19 | 9.47 | 99.36 | .. | .. | ..... |
| 21 | 8.57 | 99.54 | 66 | 2.73 | 99.99 |
| 23 | 7.83 | 99.64 | 67 | 2.69 | 100.00 |

Table 4.2: PCA compression of movements. The compression ratio and the variance proportion of the PCA components, trained from 33032, $10 \times 18$-dimensional movements of the S1–S3 HumanEva subjects.

### 4.5.2 Motion generation

The large variety of human motion suggests that a single motion model is not enough for replicating arbitrary motion. Given the current movement, m, with bfv$^*$ the last known BFV, algorithm 5 predicts the next BFV with one out of the four following motion modes:

- **Pose based** (**pose**): if the current movement's cluster is known (returned by `GetMC`,

line 5) then the learnt Gaussian model of the NBFV, with the mean ($\mathcal{C}_c.NBFV.\mu$) and variance ($\mathcal{C}_c.NBFV.\mathbf{P}$) generates the new BFV (line 14). This mode allows prediction with known and accurate previous movement, if MCM was trained with similar data. It is limited to prediction of poses that are the same as, or similar, to training data.

- **Random pose based** (**randompose**): a random jump (line 7) to an arbitrary MC models sudden large pose changes, that generates the new BFV (line 14) with a learnt Gaussian model of the NBFV (as above). This mode allows instantaneous large transitions to dissimilar poses. It fails to predict unseen data and generates discontinuous motion.

- **Pose speed based** (**speed**): with the current movement's cluster (`GetMC`, line 5), the learnt Gaussian model of the *change* of the NBFV, with the mean ($\mathcal{C}_c.Speed.\mu$) and variance ($\mathcal{C}_c.Speed.\mathbf{P}$) generates the new BFV (lines 18–19). Since the model assumes the learnt speed of the parameters, it is useful if parameter changes are smooth.

- **Normal drift** (**normal**): a 0GM, with white Gaussian noise, with a variance of the $\mathcal{M}.BFV.\mathbf{P}$ learnt from the complete training data, alters the current BFV. This mode works with static or slowly changing, smooth motion and fails with large changes. It has no memory, therefore it does not use movements of the MCM, but estimates frame-by-frame each new pose from the previous

The stochastic constants of the normal distributions for the four motion modes: $\sigma_P$ and $\sigma_L$ (**pose** and **randompose**), $\sigma_S$ (**speed**), $\sigma_N$ (**normal**) are empirically set and in section 5.4.5 their effect on the tracking will be evaluated. The same constants are used for the **pose** and **randompose** modes since only the cluster that defines the transition differs. In lines 9–13, depending whether the BFV is a complete PV model ($\mathcal{M}_1$) or a partition only, $\sigma_P$ (pose) and $\sigma_L$ (limb) allow distinct variances for the two.

This motion generation is similar to Sidenbladh *et al.* [122], in the sense that both generate a new pose completing the previous pattern (*i.e.* movement) with one new pose. However, the MCM is a compact model and in contrast to [122] does not search the whole training data, but instead uses an explicit model. The Gaussian state and transition allows unseen data. On the other hand [122] is more accurate if memory usage is not critical and

**Algorithm 5**: GetNextBFV – generates next BFV from the current movement

**Input**: $\mathcal{M}$ – MC model
*mode* – motion mode
m – current movement
**Output**: bfv – new BFV

```
1  bfv* = ₀m ;                                  // get the last BFV of the movement
2  switch mode do
3      case pose, randompose:
4          if mode = pose then
5              c = GetMC(m)
6          else
7              c = U[1; n_C]
8          end
9          if M = M_1 then                      // if model is pose or limb update
10             σ = σ_P
11         else
12             σ = σ_L
13         end
14         bfv ~ N(bfv*; M.C_c.NBFV.μ, σ · M.C_c.NBFV.P)
15     end
16     case speed
17         c = GetMC(m)
18         w ~ N(0; M.C_c.Speed.μ, σ_S · M.C_c.Speed.P)
19         bfv = bfv* + w
20     end
21     case normal
22         bfv ~ N(bfv*; 0, σ_N · M.BFV.P)
23     end
24 end
```

the pattern matches the training data well.

### 4.5.3 Model learning

Summarising the above mechanisms, the trained model $\mathcal{M}$ consists of:

- $l_m$, length of movements,
- $n_{\mathcal{C}}$, number of clusters,
- $n_{pca}$, number of PCA components used for model compression,
- BT, movement projection vector onto the PCA space,
- $M.\mu$, movement mean value,

- $BFV.\mathbf{P}$, global covariance of the BFVs,
- and for each cluster $\mathcal{C}_i$:
  - $Prior$, prior frequency of the cluster,
  - $PCA.\mu$, mean of the compressed movements,
  - $PCA.\mathbf{P}$, covariance of the compressed movements,
  - $NBFV.\mu$,mean of the NBFVs,
  - $NBFV.\mathbf{P}$, covariance of the NBFVs,
  - $Speed.\mu$, mean of the speed of NBFVs and
  - $Speed.\mathbf{P}$, covariance of the speed of NBFVs.

**Algorithm 6**: MCM learning

**Input**: $\mathrm{S} = \bigcup_{i=sequences} \bigcup_{t=poses} \mathrm{bfv}_{i,t}$ – full pose sequence database
$l_m$ – length of movements
$n_{\mathcal{C}}$ – number of clusters to generate
$\tau$ – minimum variance of selected eigenvectors

**Output**: $\mathcal{M}$ – motion model

$\{\mathrm{m'}_k\}$ = `BuildAllValidMovements`($\{\mathrm{bfv}_{i,t}\}$, $l_m + 1$)
$\{\mathrm{m}^*_k\}$=`Normalisation`($\{\mathrm{m}'_k\}$) `// normalise`
$[n^*_{pca}$ BT $M.\mu^*]$ = `PCA`($\{\mathrm{m}^*{}_k\}$, $\tau$) `// compute PCA decomposition with energy` $\tau$
$[\{\mathcal{C}_{m_k}\}, n_{\mathcal{C}}]$ = `Cluster`($\{\mathrm{pca}_{\mathrm{m}^*_k}\}$, $n_{\mathcal{C}}$) `// cluster with EM into` $n_{\mathcal{C}}$ `clusters`
$\{\mathrm{m}_k\}$ = `BuildAllValidMovements`($\{\mathrm{bfv}_{i,t}\}$, $l_m$)
$\{\mathrm{nbfv}_k\} = \{\mathrm{m}_{k+l_m}\}$
$[\mathcal{M}.n_{pca}$ $\mathcal{M}$.BT $\mathcal{M}.M.\mu]$ = `PCA`($\{\mathrm{m}_k\}$, $n_e$) `// compute PCA decomposition with energy` $\tau$
$\mathcal{M}.n_{\mathcal{C}} = n_{\mathcal{C}}$
$\mathcal{M}.BFV.\mathbf{P} = \mathrm{cov} < \{\mathrm{bfv}_{i,k}\} >$
**foreach** $\mathcal{C}_c$, $c = 1..n_{\mathcal{C}}$ **do**
    $\mathcal{M}.\mathcal{C}_c.Prior = \frac{|\{\mathrm{m}_k\}_{\mathcal{C}_{\mathrm{m}_k}=c}|}{|\{\mathrm{m}_k\}|}$ `// Cluster prior`
    $\mathcal{M}.\mathcal{C}_c.PCA.\mu = \mathcal{E} < \{\mathrm{pca}_{\mathrm{m}k}\}_{\mathcal{C}_{\mathrm{m}_k}=c} >$ `// PCA space mean,`
    $\mathcal{M}.\mathcal{C}_c.PCA.\mathbf{P} = \mathrm{cov} < \{\mathrm{pca}_{\mathrm{m}k}\}_{\mathcal{C}_{\mathrm{m}_k}=c} >$ `// and covariance`
    $\mathcal{M}.\mathcal{C}_c.NBFV.\mu = \mathcal{E} < \{\mathrm{nbfv}_k\}_{\mathcal{C}_{\mathrm{m}_k}=c} >$ `// Full parameter space`
    $\mathcal{M}.\mathcal{C}_c.NBFV.\mathbf{P} = \mathrm{cov} < \{\mathrm{nbfv}_k\}_{\mathcal{C}_{\mathrm{m}_k}=c} >$ `// and covariance`
    $\mathcal{M}.\mathcal{C}_c.Speed.\mu = \mathcal{E} < \{\mathrm{nbfv}_k - \mathrm{nbfv}_{k-1}\}_{\mathcal{C}_{\mathrm{m}_k}=c} >$ `// mean of difference`
    $\mathcal{M}.\mathcal{C}_c.Speed.\mathbf{P} = \mathrm{cov} < \{\mathrm{nbfv}_k - \mathrm{nbfv}_{k-1}\}_{\mathcal{C}_{\mathrm{m}_k}=c} >$ `// and covariance of movements`
**end**

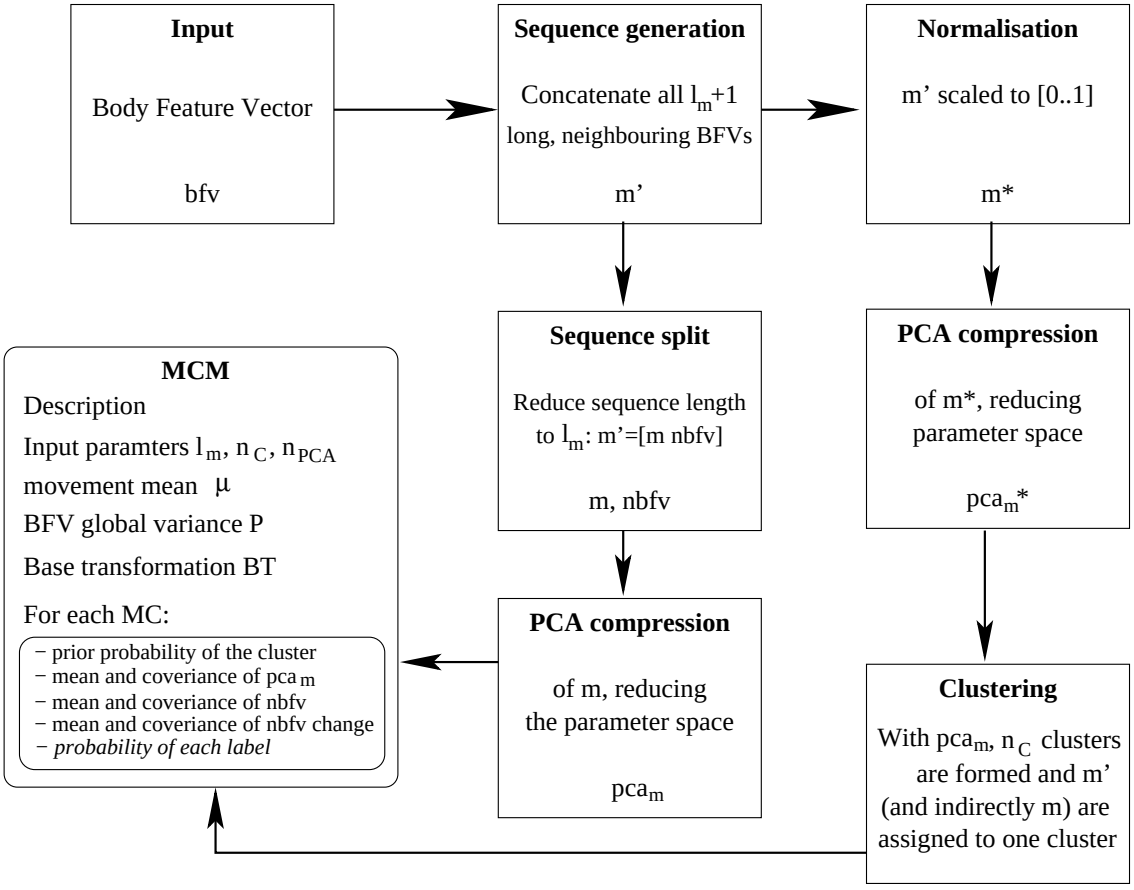



Figure 4.12: Block diagram of the MCM learning algorithm. The set of input bfv are transformed by a series of transformation, clustered to form MCs, and for each cluster the statistical properties are learnt, resulting in the MCM. Action labels for each cluster are attached later in this chapter.

This structure is the output of the unsupervised learning from algorithm 6 with the block diagram from figure 4.12. Apart from the training BFV sequences $\text{bfv}_{i,k}$, the input model parameters are the length $l_m$, of a movement, the number $n_{\mathcal{C}}$, of the desired MCs and the compression factor $\tau$ of the model.

First, `BuildAllValidMovements` concatenates BFVs in movements of length $l_m + 1$ (figure 4.12, *Sequence generation*). `Normalisation` (line 2) scales the nominal range of the parameters, defined by physical constraints of the joint angles, into $[0, 1]$ that allows in the clustering equal importance to all parameters. Next, PCA reduces the dimensionality of highly correlated movements (lines 3).

The MC are generated by clustering the $l_m$ +1 long, compressed movements (line 4). Then, similarly to the above, the PCA compression (line 7) and the succeeding BFV are found (line 6) for the $l_m$ long movements. These are clustered into one of the MCs, and in lines 8–18, for each cluster, Gaussian models are trained for the model $\mathcal{M}$ features defined earlier.

The BFVs available to train the motion model are subsets of the articulated human model PVs, defined in chapter 3. With different levels of detail, table 4.3 defines 14 models ($\mathcal{M}_i$) trained with: the full articulated body joint angle BFV (*i.e.* Whole body), the complete (Head, Left/Right Arm/Leg), the lower and upper limbs MCMs.

| **Model / Level** | **Description** | **Parameter partitions ($\phi$)** |
|---|---|---|
| $\mathcal{M}_1$ | Whole body | 5..6, 9..24 |
| $\mathcal{M}_2$ | Head | 7..8 |
| $\mathcal{M}_3$ | Left Arm | 9..12 |
| $\mathcal{M}_4$ | Right Arm | 13..16 |
| $\mathcal{M}_5$ | Left Leg | 17..20 |
| $\mathcal{M}_6$ | Right Leg | 21..24 |
| $\mathcal{M}_7$ | Left Upper Arm | 9..10 |
| $\mathcal{M}_8$ | Right Upper Arm | 13..14 |
| $\mathcal{M}_9$ | Left Upper Leg | 17..18 |
| $\mathcal{M}_{10}$ | Right Upper Leg | 21..22 |
| $\mathcal{M}_{11}$ | Left Lower Arm | 11..12 |
| $\mathcal{M}_{12}$ | Right Lower Arm | 15..16 |
| $\mathcal{M}_{13}$ | Left Lower Leg | 19..20 |
| $\mathcal{M}_{14}$ | Right Lower Leg | 22..24 |

Table 4.3: The set of MCMs. Partitions $\text{p}_\phi$ of the pose p generate multiple BFV sets that result in an $\mathcal{M}_i$ MCM. Each MCM has different complexity and refers to one or more body parts.

During algorithm testing, it was observed that the head parameters are unstable, both in the training and tracking data. Therefore $\mathcal{M}_2$ is not used in prediction or recognition.

### 4.5.4 Movement likelihood and movement conditioned MC probability

The MC probabilistic membership of an arbitrary movement results from the statistical modelling above. Next, in order to simplify the notation for an arbitrary model, the model $\mathcal{M}$ will be implicit.

The prior probability of the cluster $\mathcal{C}_i$ is

$$\mathcal{P}(\mathcal{C}_i) = \mathcal{C}_i.Prior. \tag{4.13}$$

With PCA compression the probability of a movement $m$ conditioned by the MC is expressed by the probability of the compressed representation:

$$\mathcal{P}(\mathrm{m}|\mathcal{C}_i) = \mathcal{P}(\mathrm{pca_m}|\mathcal{C}_i). \tag{4.14}$$

A normal probability density function models the cluster of the compressed movements, therefore

$$\mathcal{P}(\mathrm{m}|\mathcal{C}_i) = c_1 \cdot e^{\delta^T_{\mathcal{C}_c}(\mathrm{pca_m})\cdot\mathcal{C}_c.PCA.\mathbf{P}^{-1}\cdot\delta_{\mathcal{C}_c}(\mathrm{pca_m})}, \tag{4.15}$$

where $\delta_{\mathcal{C}_c}(\mathrm{pca_m}) = \mathrm{pca_m} - \mathcal{C}_c.PCA.\mu$.

Further, the probability of a movement m being cluster $\mathcal{C}_c$ using Bayes rule is

$$\begin{aligned} \mathtt{Sim}_{\mathcal{C}_c}(\mathrm{m}) &= \mathcal{P}(\mathcal{C}_c|\mathrm{m}) \\ &= \frac{\mathcal{P}(\mathrm{m}|\mathcal{C}_c)\ \mathcal{P}(\mathcal{C}_c)}{\mathcal{P}(\mathrm{m})} \\ &= c_2 \cdot \mathcal{C}_i.Prior \cdot e^{\delta^T_{\mathcal{C}_c}(\mathrm{pca_{ap}})\cdot\mathcal{C}_c.PCA.\mathbf{P}^{-1}\cdot\delta_{\mathcal{C}_c}(\mathrm{pca_{ap}})}. \end{aligned} \tag{4.16}$$

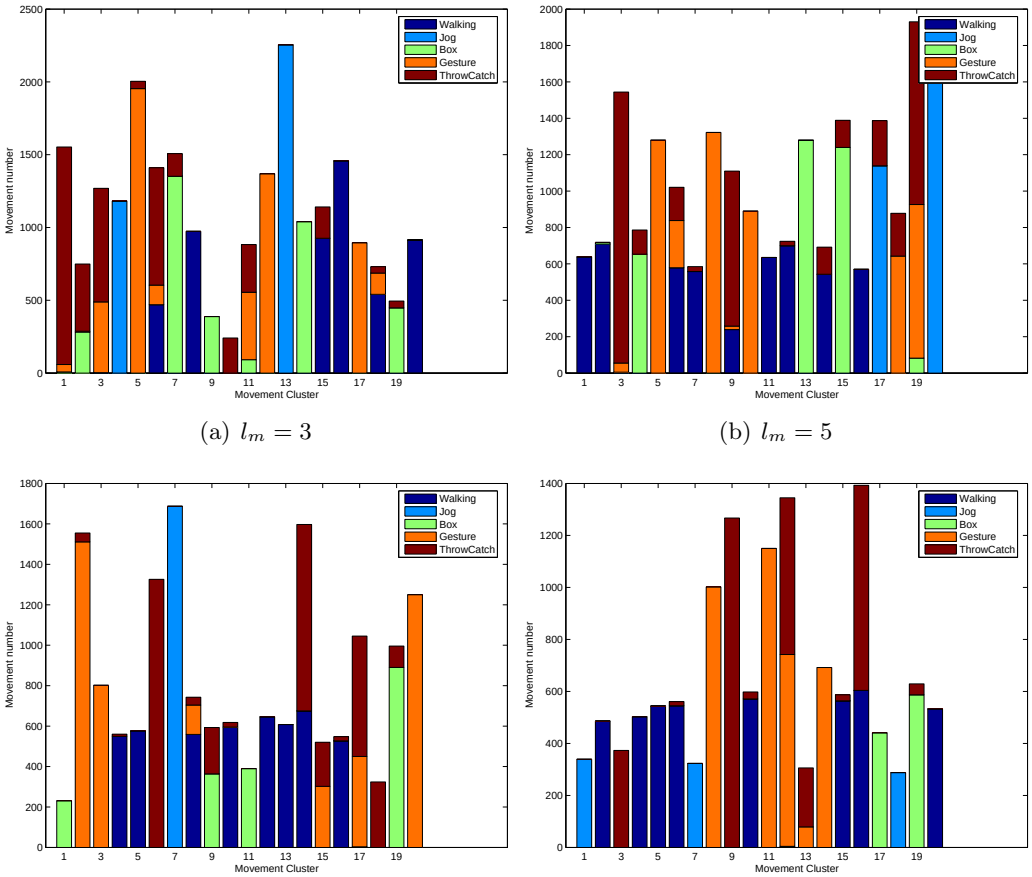
Finally, with maximum a posteriori, the MC of an arbitrary movement m is the most similar cluster:

$$\mathtt{GetMC}(\mathrm{m}) = \underset{\mathcal{C}_c}{\operatorname{argmax}}\, \mathtt{Sim}_{\mathcal{C}_c}(\mathrm{m}). \tag{4.17}$$

### 4.5.5 Experiment: MC uniformity

In this experiment, the effect of $l_m$ and $n_{\mathcal{C}}$ on composition of the MC is examined. The model $\mathcal{M}_1$ (*i.e.* BFV with the 18 whole body parameters) was trained with all valid movements from the *train* partition of the HumanEva dataset [194], a total of 10833 ($l_m = 35$) to 23377 ($l_m = 1$) movements, with subjects S1 and S2 only, for the combinations of $n_{\mathcal{C}} = 20, 40, 60, 80$ and 100 clusters and $l_m = 1, 3, 5, 15, 25$ and 35 long movements.

Diagrams from figures 4.13 and 4.14 show the distribution of all training movements from the five categories of training sequences. For a good classification, sequences from different activities are expected to fall in different clusters, while clusters have similar movements of the same activity.




(a) $l_m = 3$ (b) $l_m = 5$

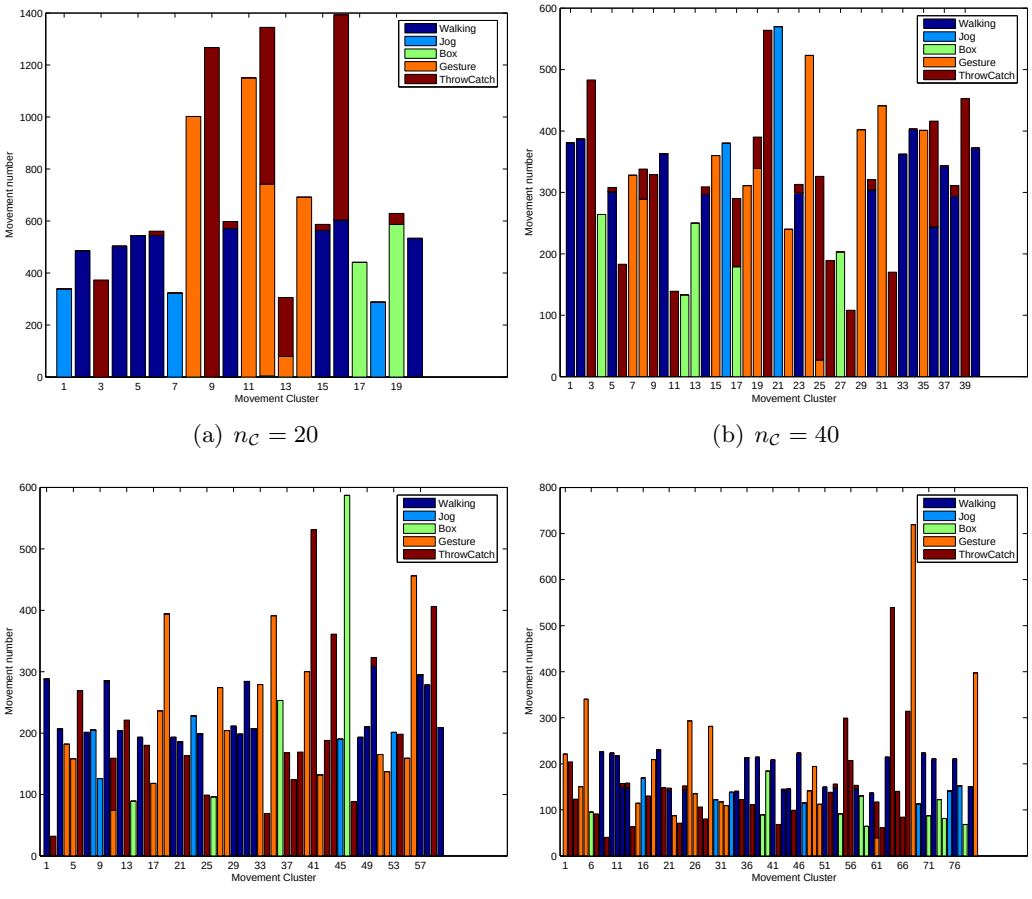
(c) $l_m = 15$ (d) $l_m = 25$

Figure 4.13: Movement length dependent cluster composition for the HumanEva basic activities, $n_{\mathcal{C}} = 20$ clusters and varying movement length $l_m$. The number of clusters with a mix of three activities reduces from 3 in (a) and (b), to 1 in (c) and 0 in (d)

In figure 4.13(a) MC no. 11 has in total 883 movements from *Box*, *Throw/Catch*

and *Jog* sequences. This cluster is the worst in figure 4.13a, since the membership of a movement in a cluster does not constrain the activity of which the motion is a member. This artefact is motivated by the similarity of *Box*, *Throw/Catch* and *Jog*. For longer movements, greater $l_m$, clusters have more dominant activities. This concludes that for $n_{\mathcal{C}} = 20$ clusters the increase of $l_m$ results in better classification, since over a longer observation the classification is more stable.

Figure 4.14 shows the MC composition for $l_m = 25$ long movements for several cluster numbers. It suggests that with the increase of $n_{\mathcal{C}}$, the clusters with larger covariance are split and they are more discriminatory.




(a) $n_{\mathcal{C}} = 20$ (b) $n_{\mathcal{C}} = 40$

(c) $n_{\mathcal{C}} = 60$ (d) $n_{\mathcal{C}} = 80$

Figure 4.14: MC number dependent cluster composition for the HumanEva basic activities, $l_m = 25$ movement length and varying cluster number $n_{\mathcal{C}}$.

The above histograms' visual comparison is subjective. For objectivity, a measure of

uniformity was defined as

$$u = \sum_{\chi \in X} \frac{(c_\chi^{max})^2}{\sum_{\alpha \in A(\chi)} c_{\chi,\alpha}} / \sum_{\chi \in X} c_\chi^{max}, \tag{4.18}$$

with

$$c_\chi^{max} = \max_{\alpha \in A(\chi)} (c_{\chi,\alpha}), \tag{4.19}$$

where $X$ is the set of MCs, $A(\chi)$ is the set of activities of cluster $\chi$, and $c_{\chi,\alpha}$ is the histogram value of movements of activity $\alpha$ in bin (*i.e.* cluster) $\chi$. The uniformity $u$ is one if all clusters have movements from a single cluster only; otherwise it favours clusters with a higher number of movements, penalising those with fewer. The minimum value is $1/|A|$, the inverse of the cardinality of the activity set, *e.g.* the minimum uniformity for the above 5 activities is 20%. This minimum results for the case that all MCs, equal in number, are in a single bin.

| $n_\mathcal{C}$ | $l_m$ | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 3 | 5 | 15 | 25 | 35 |
| 20 | 91.51% | 91.72% | 90.43% | 91.52% | 93.01% | 95.70% |
| 40 | 94.61% | 93.60% | 94.85% | 94.53% | 97.31% | 97.41% |
| 60 | 96.15% | 95.72% | 96.73% | 96.91% | 99.53% | 99.08% |
| 80 | 96.44% | 96.84% | 97.56% | 98.54% | 99.29% | 98.96% |
| 100 | 96.49% | 97.40% | 97.89% | 98.79% | 99.16% | 99.39% |

Table 4.4: Cluster uniformity $u$, in relation to the number of clusters $n_\mathcal{C}$ and the sequence length $l_m$

Table 4.4 confirms that conclusion already drawn that increases in both the movement length and the number of clusters enhance the MC uniformity. The above uniformity is maximal for $n_\mathcal{C} = 60$ and $l_m = 25$. However, it is not a perfect metric, it reflects only on activity level and have has no fine scale behaviours such as defined in section 4.6. Also, as next section will show, the MC can be used for pose prediction, of which quality will be dependent on $n_\mathcal{C}$ and $l_m$. This will be subject of the tracker analysis in section 5.4.1. Therefore, here it can be concluded only that, as desired, MCs contain similar movements and the discrimination between global actions enhances with more clusters and longer movements.

One would expect the number of clusters to be equal to the number of activities (*i.e.* five for HumanEva). This is not the case for many reasons: the high dimensional parameter space is multi-modal, and clusters are used to classify not just one cluster of

exclusive activities, but also actions, which can overlap and combine independently with other actions. This requires a $n_{\mathcal{C}}$ high enough to allow combinations between different actions. Adding more clusters is limited by the training set size, because each cluster requires to train its mean and covariance, a number of member movements more than the dimensionality of the parameter space.

### 4.5.6 Synthetic motion generation

Synthetic random motion with a *model*, $\mathcal{M}_1 \dots \mathcal{M}_{14}$, is summarised in algorithm 7. This algorithm generates poses with motion $mode = pose$, by calling the `GetNextBFV` from algorithm 5. Since the movement initialisation is not straightforward, the artefact used in line 3 sets all BFVs from the previous movement equal to the NBFV of the starting cluster. This may cause the first MC to differ from $mc_0$, however this is irrelevant to verifying the visual correctness of the motion sequences.

**Algorithm 7**: MCM synthetic motion generator

**Input**: $\mathcal{M}$ – MCM partition
$mc_0$ – initial MC
**Output**: $\{\text{bfv}_t\}$ – sequence of generated BFV

1 $mode = pose$
2 $t = 0$
3 $\text{bfv}_{-l_m+1\dots0} = \mathcal{M}.\mathcal{C}_{mc_0}.NBFV.\mu$ `// initial BFVs`
4 **repeat**
5 $\quad \text{m} = [\text{bfv}_{t-l_m+1} \dots \text{bfv}_t]$ `// current movement`
6 $\quad t = t + 1$
7 $\quad \text{bfv}_t =$ `GetNextBFV(` $\mathcal{M}$, *mode*, m`)` `// generate new BFV`
8 **until** *enough*

The synthetic motion from algorithm 7 is expected to generate a sequence of physically valid poses, in a valid temporal sequence. The first test (figure 4.15) generates left whole leg BFVs ($\mathcal{M}_5$), while the second (figure 4.17) whole body parameters ($\mathcal{M}_1$). The stochastic constants are $\sigma_P = 0.5$ for the pose and $\sigma_L = 1$ for the limb model. The MCM used has $n_{\mathcal{C}} = 100$ and $l_m = 3$.

Figure 4.16 shows the MC transitions. Transitions to the same MC are present, since consecutive poses are similar, however the stochastic component of the MC ensures that the poses are not identical. The graph has loop switch alternative parallel branches, suggesting alternative dynamics.

Figure 4.17 and 4.18 show the synthetic motion generated for the whole body. The

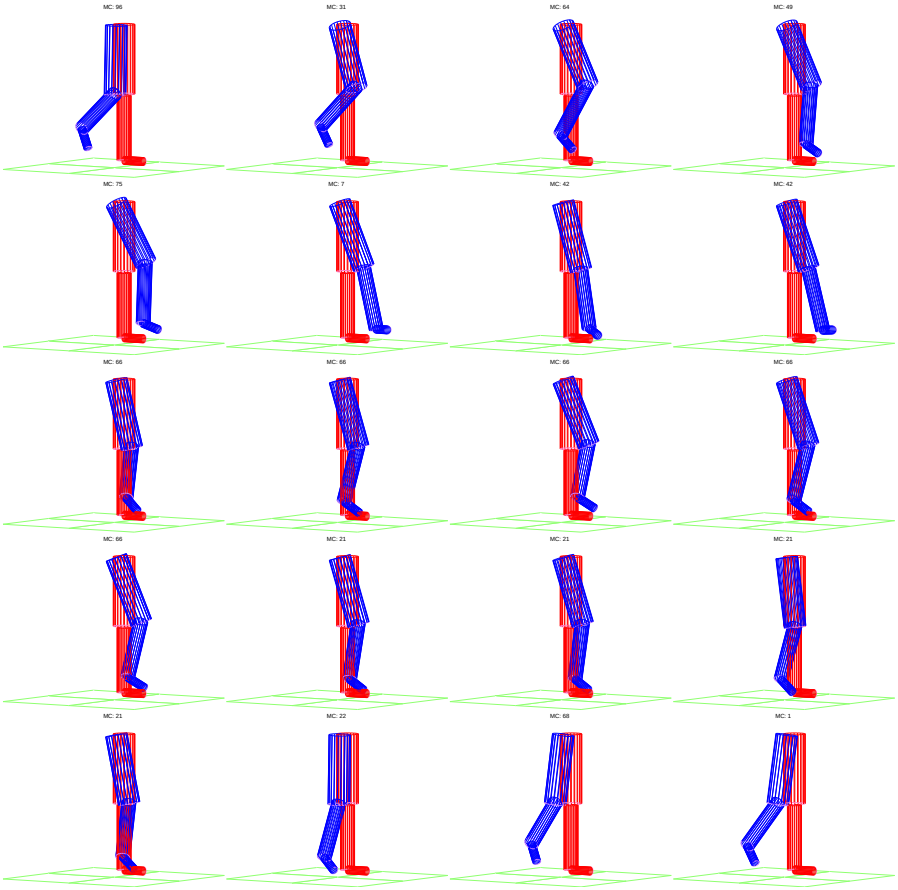



Figure 4.15: Left leg random motion ($mode = pose$ only) with MCM $Model_5$. $n_{\mathcal{C}} = 100$, $l_m = 3$. The initial MC is $mc_0 = 1$, not shown in the figure [◊].

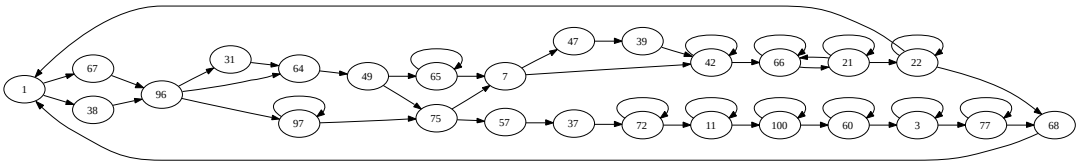



Figure 4.16: A leg MC transition sequence. The graph of 100 randomly generated transitions between leg MCs, with the first 20 poses shown in figure 4.15.

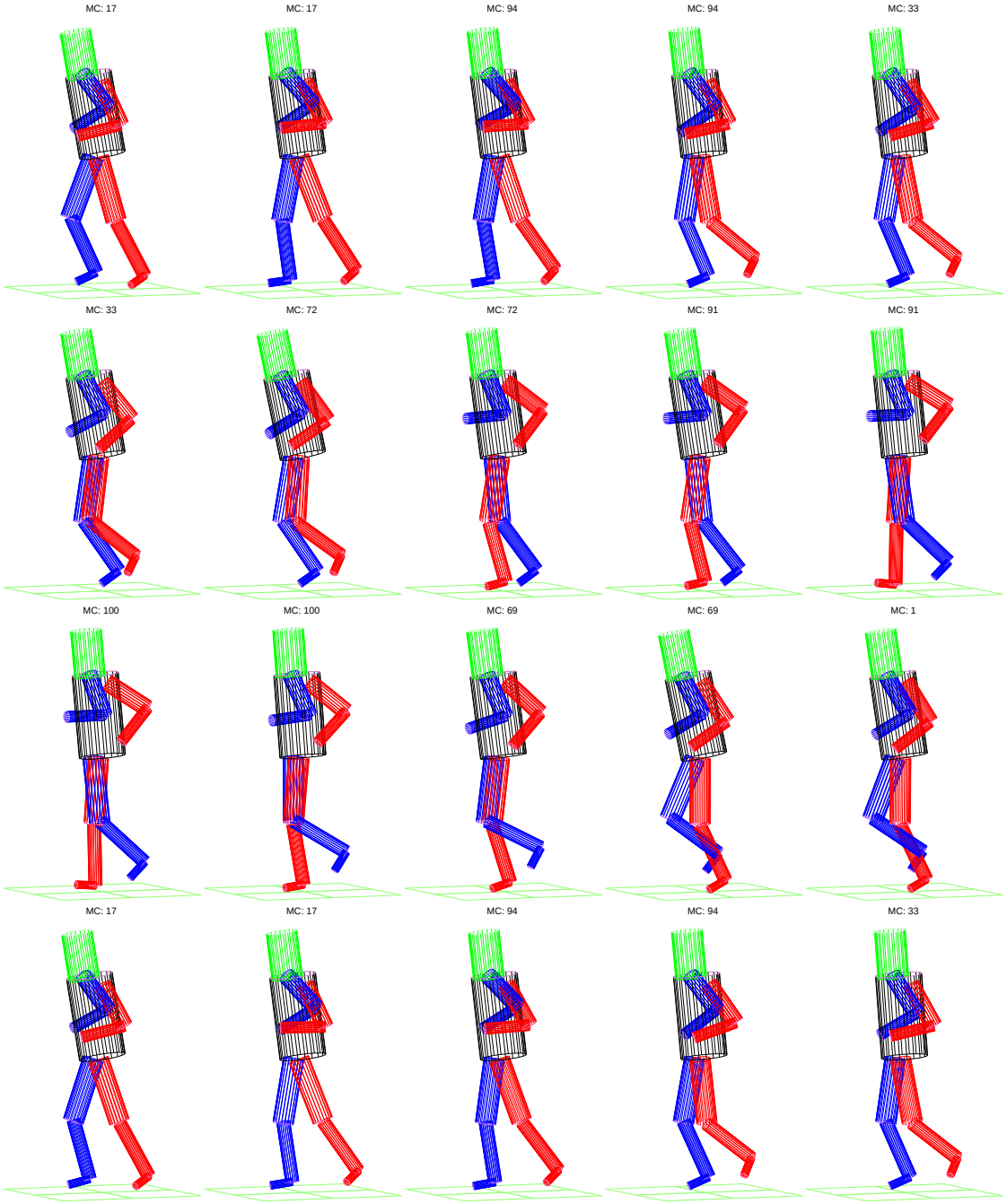



Figure 4.17: Full body ($mode = pose$ only) with MCM $Model_1$. $n_{\mathcal{C}} = 100$, $l_m = 3$. The initial MC is $mc_0 = 1$, not shown in the figure [◊].

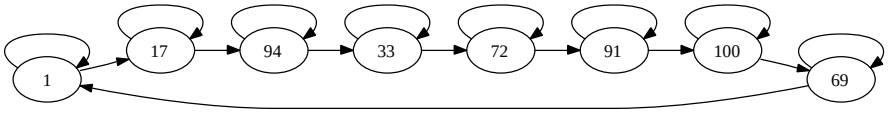



Figure 4.18: A whole-pose MC transition sequence. The graph of 100 randomly generated transitions between body MCs, with the first 20 poses shown in figure 4.17.

period of a walking cycle contains eight MC. As both figures show, self transitions occur, however the poses generated are different. In this example, the motion does not stay stationary in any of the MC for longer than two poses.

Figure 4.19 expands figure 4.18 and shows three MCs transition sequences with 2000 poses generated with $\mathcal{M}_1$. Loops of the graph correspond to periodic motion, while the multiple loops suggest different activities and a very wide range of possible poses. The figures 4.19(a) and 4.19(b) both have $mc_0 = 1$ starting MC, however the stochastic generator results in different pose sequences, *i.e.* through $mc = 17$ the sequence changes to a different periodic pose sequence, possibly part of a different activity. Figure 4.19(c) has different starting MC, and therefore the pose sequence is completely changed.

## 4.6 Behaviour primitives

The MCM defines similar movements by means of the MCs. Therefore, in section 4.1 it was suggested that semantic activity labels can be attached to MC.

Since movements are defined over a duration, it is important to specify the time a movement label refers to. Here, the label of a movement is given by the *last* frame, *i.e.* the actual label is defined by the previous and current configuration.

A label is a semantic description of the movement and is an action or activity name. Since complex activities require a temporal combination of multiple actions with an extent, then to reduce the complexity, activities are defined in this thesis by one or a set of independent actions, and activity recognition becomes action recognition. The independent recovery of the trained action labels will provide detailed information about the activity, an activity description beyond simple classification into a set of limited actions. As result, an activity might have multiple labels, such as *Standing* and *Pointing (a gun)* that could signal a dangerous situation, or *Walking*, *Head looking down*, *Arms straight* and *Arms behind* that could describe a "human in a thinking process" while walking with arm held behind the body.

The model is based on the idea that actions can be partly or completely constructed from smaller components, the movements. A movement can have multiple labels, being part of one or more actions. For an arbitrary movement, the resembling MC provides the probability of each action label.

Only the direct BFVs are useful for the motion prediction. However, other BFV that

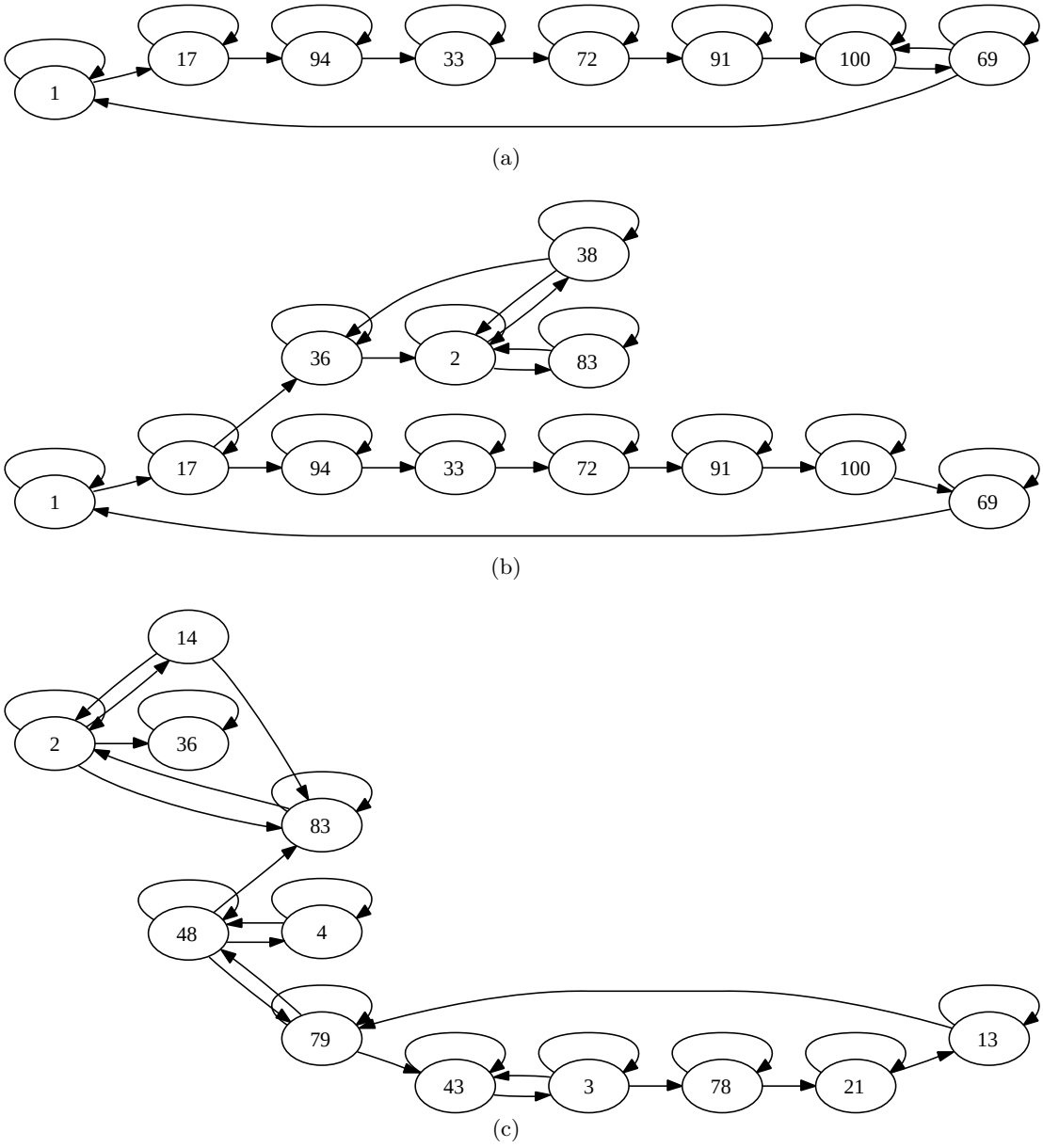



Figure 4.19: Whole-pose MC transition sequences for 2000 MCs. (a) and (b) have the same MC initialisation, and hence comparable sequences, while (c) has different initialisation.

include derived features such as speed and radial coordinates can be used for recognition. The subsequent analysis is not limited to direct features. Further, since the same MCMs are used both for prediction and recognition, this thesis analyses the recognition with direct BFV only.

### 4.6.1 Action learning

The MCM are built with algorithm 6 and, as a supplementary training phase, symbolic labels are attached to each cluster. For this, the supervised training requires that each instance, m, is labelled with the appropriate set of labels. The label presence, $L_l(\mathrm{m})$, is one if movement m is labelled with $l$, and zero if not.

The probability of a label $l$ conditioned by a movement cluster, $\mathcal{C}_i$ is the frequency of the label weighted by the movement similarity with the cluster:

$$\mathcal{P}(l|\mathcal{C}_i) = \frac{\sum_{\mathrm{m}} \mathtt{Sim}_{\mathcal{C}_i}(\mathrm{m}) \cdot L_l(\mathrm{m})}{\sum_{\mathrm{m}} \mathtt{Sim}_{\mathcal{C}_i}(\mathrm{m})}. \tag{4.20}$$

The probability of a label, given the current movement m, is the integrated marginal probabilities over all clusters:

$$\begin{aligned} \mathcal{L}_l(\mathrm{m}) &= \mathcal{P}(l|\mathrm{m}) \\ &= \sum_{\mathcal{C}_i} \mathcal{P}(l, \mathcal{C}_i|\mathrm{m}) \\ &= \sum_{\mathcal{C}_i} \mathcal{P}(l|\mathcal{C}_i, \mathrm{m})\mathcal{P}(\mathcal{C}_i|\mathrm{m}) \\ &= \sum_{\mathcal{C}_i} \mathcal{P}(l|\mathcal{C}_i)\mathcal{P}(\mathcal{C}|\mathrm{m}), \end{aligned} \tag{4.21}$$

with $Pr(l|\mathcal{C}, \mathrm{m}) = \mathcal{P}(l|\mathcal{C})$ from the functional dependence of the cluster $\mathcal{C}$ from the movement m.

### 4.6.2 Action labels of the HumanEva dataset

The HumanEva dataset (section 2.6.1) includes sequences of five activities: *Walk*, *Jog*, *Throw/Catch*, *Box* and *Gesture*. Each movement of these sequences is therefore an action

of the corresponding activity sequence, thus each movement can be explicitly described with this label. All 20 training sequences of subject S1 and S2 were labelled with one of the *Walk*, *Throw/Catch*, *Jog*, *Gesture* and *Box* labels.

Further to the global labels, five of the HumanEva *train* sequences were labelled with the detailed labels from table 4.5.

| **Label** | **Description** | **Sequence** |
|---|---|---|
| *Left/right stride back* | Left/right leg is moving forward from behind of the right/left leg | S1 Walking 1,<br>S1 Walking 3. |
| *Left/right stride front* | Left/right leg is moving forward ahead of the right/left leg | S1 Walking 1,<br>S1 Walking 3. |
| *Left/right arm forward* | Left/right arm is moving forward | S1 Walking 1,<br>S1 Walking 3,<br>S1 ThrowCatch 1,<br>S2 ThrowCatch 1,<br>S2 ThrowCatch 3. |
| *Left/right arm backward* | Left/right arm is moving back-wards | S1 Walking 1,<br>S1 Walking 3,<br>S1 ThrowCatch 1,<br>S2 ThrowCatch 1,<br>S2 ThrowCatch 3. |
| *Left/right hand throw* | Right/left hand throw | S1 Walking 1,<br>S1 Walking 3,<br>S1 ThrowCatch 1,<br>S2 ThrowCatch 1,<br>S2 ThrowCatch 3. |

Table 4.5: Local labels with descriptions and training sequences. For each left and right side, five pairs of action labels are trained with two or five HumanEva sequences.

The global (*e.g.* *Walk*, *Throw/Catch*, *etc.*) and the local labels (*e.g.* *left stride back*, *left hand throw*) are considered at the same semantic *action* level. One can argue that global labels are activities, however components of a long (*e.g.* whole sequence) sequence can be viewed as an action that define the activity with the same name. Hence, *Walk* is also an action and it is part of the *Walk* activity. Further, the *Walk* activity could also be inferred using local labels, however this is not aimed in this thesis. Also, a *Walk* action is part of more complex activities such as *Shopping*. In this thesis, activity description is composed from concurrently detected global and local actions that provide detailed behavioural information.

For labelling, a Matlab tool was designed (figure 4.20) that allows concurrent label specification with a set of labels, and label management operations, either frame by frame or over a continuous range of frames. The interface uses a 2D view (figure 4.20(a)) or the MOCAP data visualised as a rotatable 3D pose (figure 4.20(b)). More recently, ViPER

[195] is a largely accepted tool to perform labelling of ground truth data, however 3D MOCAP poses require visualisation not compatible with ViPER.

### 4.6.3 MCM behavioural analysis of the HumanEva dataset

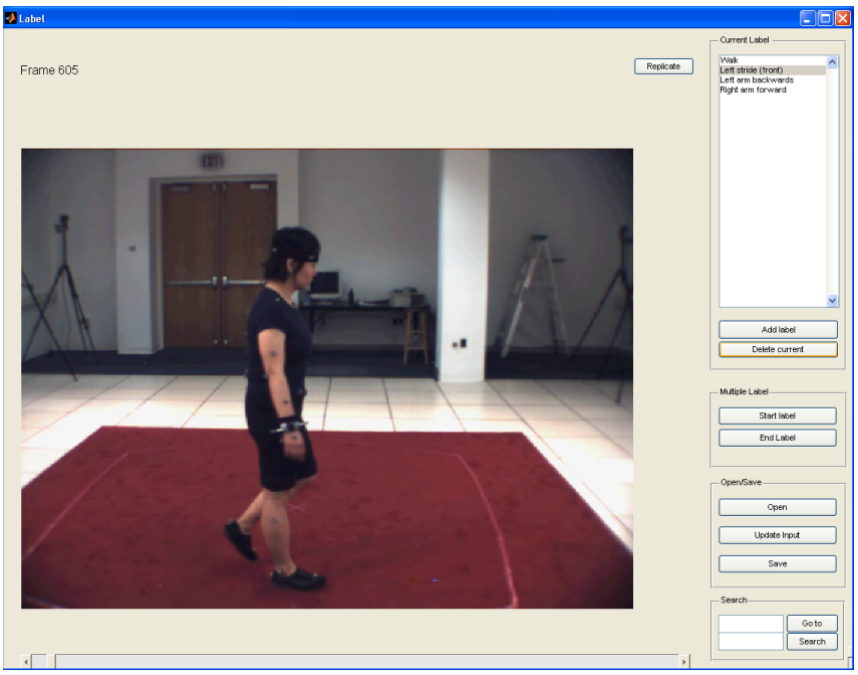
To evaluate model $\mathcal{M}_1$, trained with labels from table 4.21, the MOCAP data of the *validate* HumanEva sequences are used. While subjects S1 and S2 have training data (from the *train* sequences, distinct from *validate*), subject S3 was not included in the training, therefore these data represent the recognition for unseen subjects. First, subjects S1 and S3 sequences are analysed. For each frame, *i.e.* each movement ending at the current frame, the label probabilities resulting from equation (4.21) are shown in figure 4.21. The *Walk* label for the whole *S1 Walking 1* test sequence is perfectly recognised; the four stride labels and four arm forward and backward labels are observed with a perfect periodicity. Their periodic nature is not exploited either in training or in recognition. The least accurate recognition is for the *S3 Gesture 1* sequence, because of its similarities with the *Throw/Catch* activity.

The visual evaluation of sequence lengths $l_m$ and cluster numbers $n_{\mathcal{C}}$ of the *S1 Walking 1* sequence from figure 4.22 suggests that selection of sequence length is more important than the cluster number, and that with higher sequence length recognition of shorter actions degrades, especially for low cluster number. An increase of $n_{\mathcal{C}}$, results in finer detail. The transition between detailed labels are smoother and have intermediate probability values.

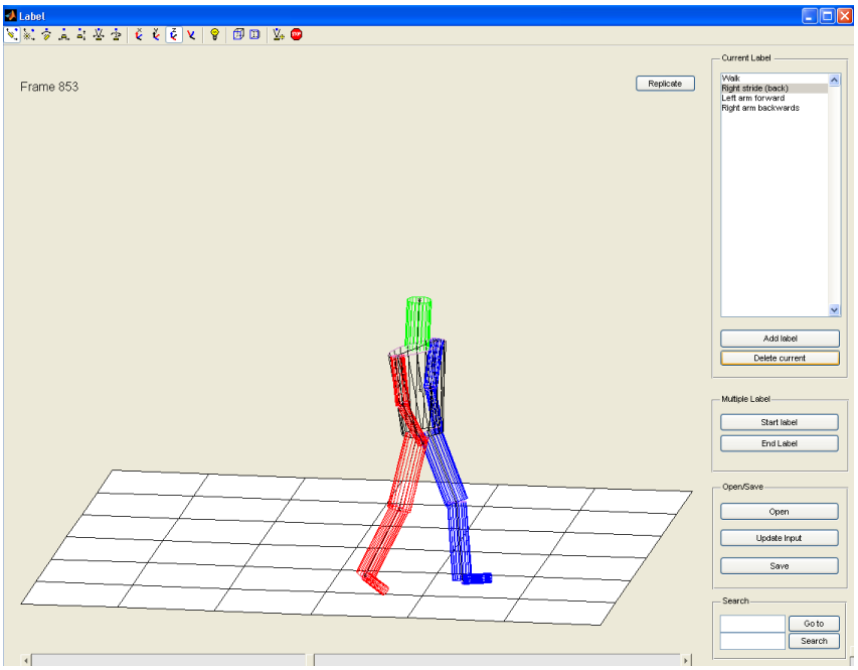
Figure 4.23 shows the confusion matrices of the global actions, defined as

$$\mathrm{C}(l_a, l_b) = \underset{\mathrm{m}\in(\text{sequences of action } l_b)}{\mathcal{E}} < \mathcal{L}_{l_a}(\mathrm{m}) > . \tag{4.22}$$

The figure represents the overall recognition performance in classifying all movements of the *validate* dataset sequences. Misclassification of *Throw/Catch*, *Gesture* and *Box* activities with *Gesture* or *Box* for $l_m = 15$, or no-detection for $(l_m = 25, n_{\mathcal{C}} = 100)$ are emphasised with longer sequence length. This is explained by the similarity of these activities, and suggests the need for analysing the data over a shorter scale. On the other hand, although activities if classified well, have stronger response, the confusion matrices with more MC do not result in better recognition, possibly because the specialised MCs are sensitive to the data that they are trained on. This effect is emphasised with longer




(a) 2D view




(b) 3D view

Figure 4.20: The labelling interface. For each frame, the user defines the set of labels attached. The labels can be *replicated* from the previous frame and *deleted* from the current frame. Multiple frames are labelled with the *start* and *end label* frames of the range. A sequence can be *opened* (created or loaded) either from an image sequence, from the MOCAP data, or from the *saved* labelled data. *Update input* allows change between the 2D and 3D visualisation. Searching for a frame number or for a label switch (*i.e.* on or off) provides additional user support.

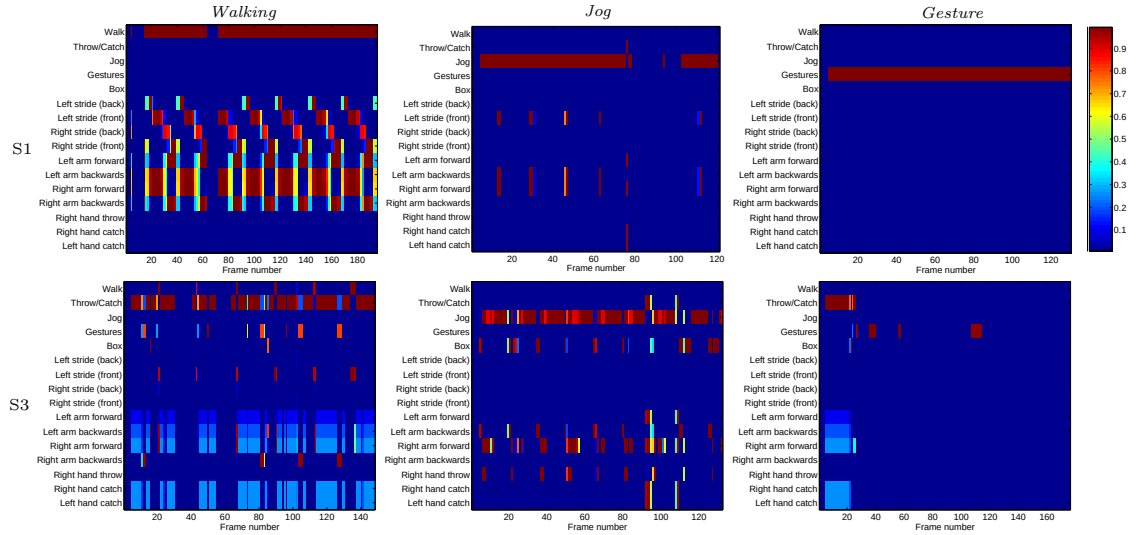



Figure 4.21: Recognition with known and unknown human subject of *Walk*, *Jog*, and *Gesture* activities. Subject S1 was trained, while S3 was not. The model has $n_{\mathcal{C}} = 100$ and $l_m = 5$. The probability of labels (vertical) for each frame (on horizontal) is colour coded. For the first $l_m - 1$ frames no movement can be defined, and for frames 6–12 and 64–69 for *Walk* S1, frames 32–40 and 56–63 for *Walk* S3, frames 80–125 for *Jog* S1, frames 28–35, 41–55, 58–106 and 115–180 for *Gesture* S3 one of $l_m$ the pose vector is missing, therefore no recognition is possible. This is visible by the vertical zero probability bands.

movements, which results in un-classified movements. Intuitively, the best values analysing only lines or columns are for $n_{\mathcal{C}} = 60$, and $l_m = 5$ and 15.

The MCM generates independent action probabilities and therefore concurrently both *Walk* and *Throw/Catch*, or even *Walk* and *Jog* can be recognised. Hence, for related actions, lines of the confusion matrices are normalised to one. Similarly, if none of the labels are recognised then all probabilities are zero, and confusion matrices have all zero lines. However, the advantage is the independent learning of the labels and extensibility with new labels.

### 4.6.4 Actions from the MCM sets

The full body BFV, *i.e.* $\mathcal{M}_1$, was employed above in the recognition. However, all 14 MCMs (table 4.3) analogously provide action labels. Figures 4.24 and 4.25 show the recognised label probabilities for the *S1 Walking 1* and *S3 Walking 1* sequences for $n_{\mathcal{C}} = 100$ and $l_m = 3$, for the 13 MCMs (all except the unreliable head MCM), and the *Overall* label with the expected label probability of the 13 MCMs.

The response of the limb level models on a label that is defined by another limb confirms the strong dependence between the parameters. However, as expected, legs, and

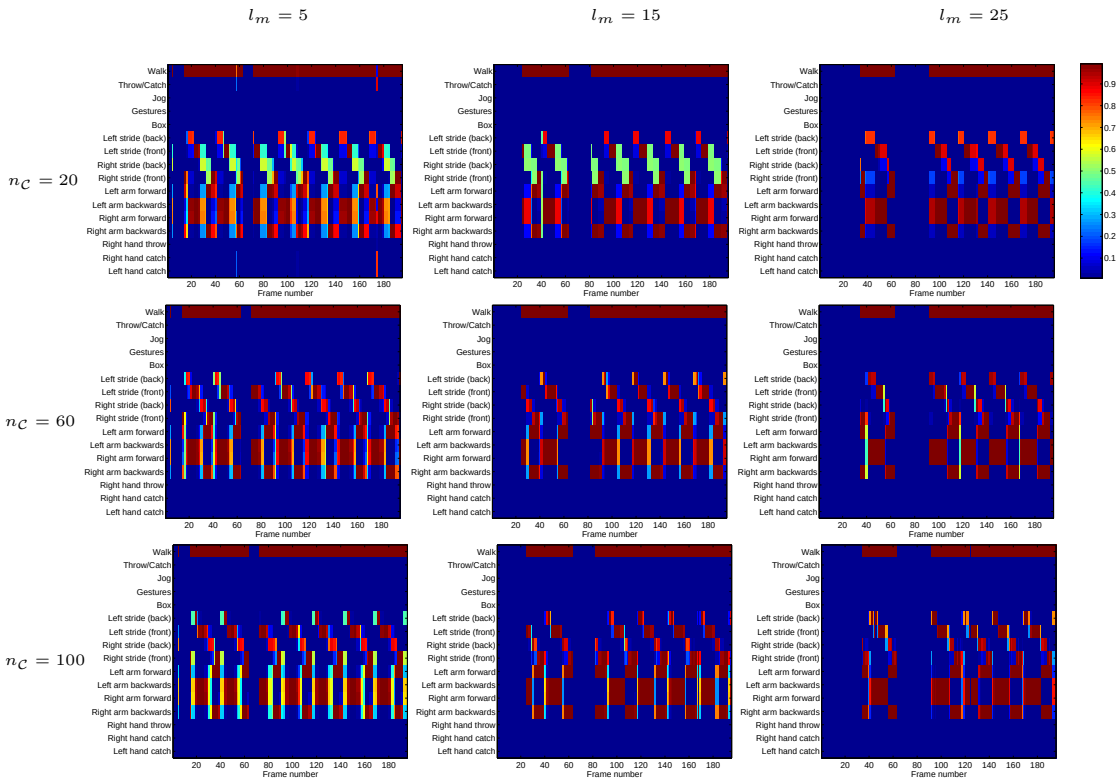



Figure 4.22: *S1 Walking 1* activity recognition for MCMs with number of clusters $n_\mathcal{C} = 20, 60, 100$ and length of sequence $l_m = 5, 15, 25$. The zero probability vertical bands result from the missing pose information for frames 6–10 and 64–67. For these and the subsequent $l_m - 1$ frames a movement cannot be defined.

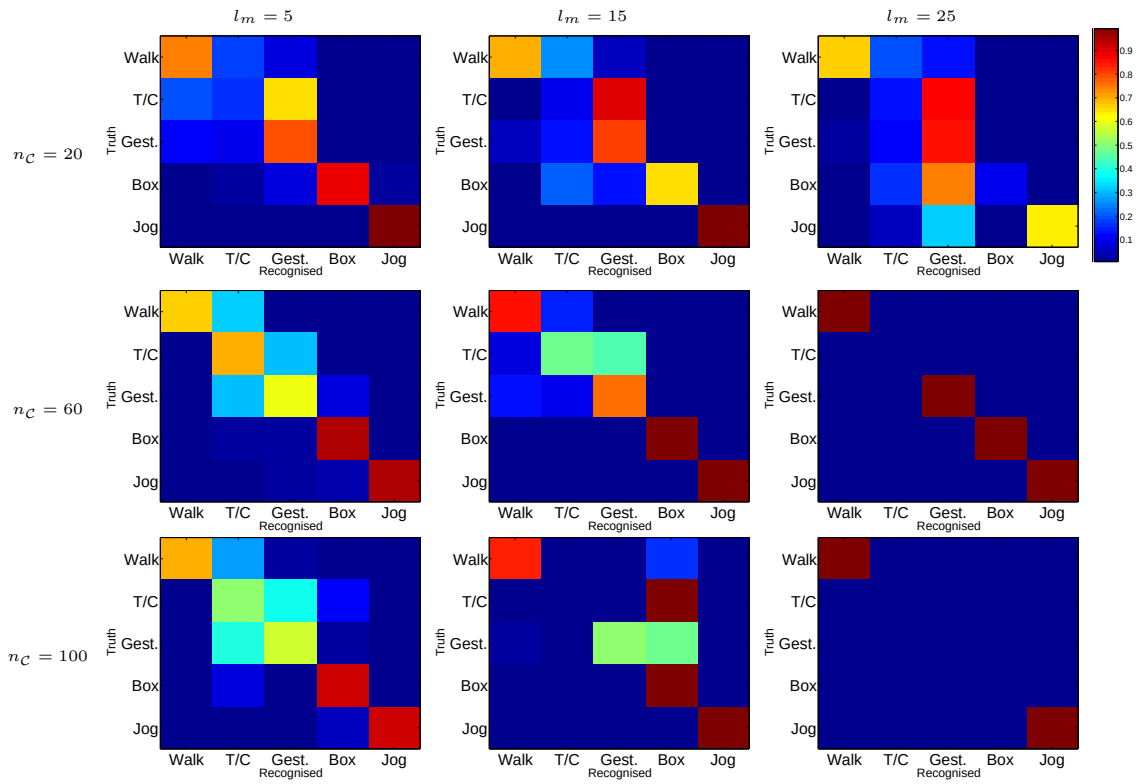



Figure 4.23: Confusion matrices for MCMs with number of clusters $n_\mathcal{C} = 20, 60, 100$ and length of sequence $l_m = 5, 15, 25$

specifically whole legs (*i.e.* $\mathcal{M}_5$ and $\mathcal{M}_6$), best resemble the periodicity of the motion. It is clear that models with fewer parameters are more specialised and provide finer probabilities of the detailed labels. Which model is more successful for action analysis is not analysed extensively, although the overall labels suggest that combining MCMs from a pool enhances label detection. However, the lower probabilities compared to individual MCM probabilities convey that some do not contribute. A study of the relevance of MCMs is outside the scope of this thesis.

### 4.6.5 Recognition evaluation

The confusion matrix (*e.g.* figure 4.23) describes the recognition success of each activity, however it is a matrix, hence is hard to compare with other confusion matrices. The accuracy $\zeta$, defined by equation (2.18) is an objective compact metric, based on the computed confusion matrix. For an $n$ class problem, and perfect recognition implies $\mathrm{C}_{i,i} = 1$, and $\mathrm{C}_{i,j} = 0$, for $i \neq j$, $1 \leq i \leq n$, therefore a $\zeta = 1$. For a random recognition $\mathrm{C}_{i,j}$ is constant and $\zeta = 1/n$. With no recognition, $\mathrm{C}_{i,i} = 0$ and $\zeta = 0$.

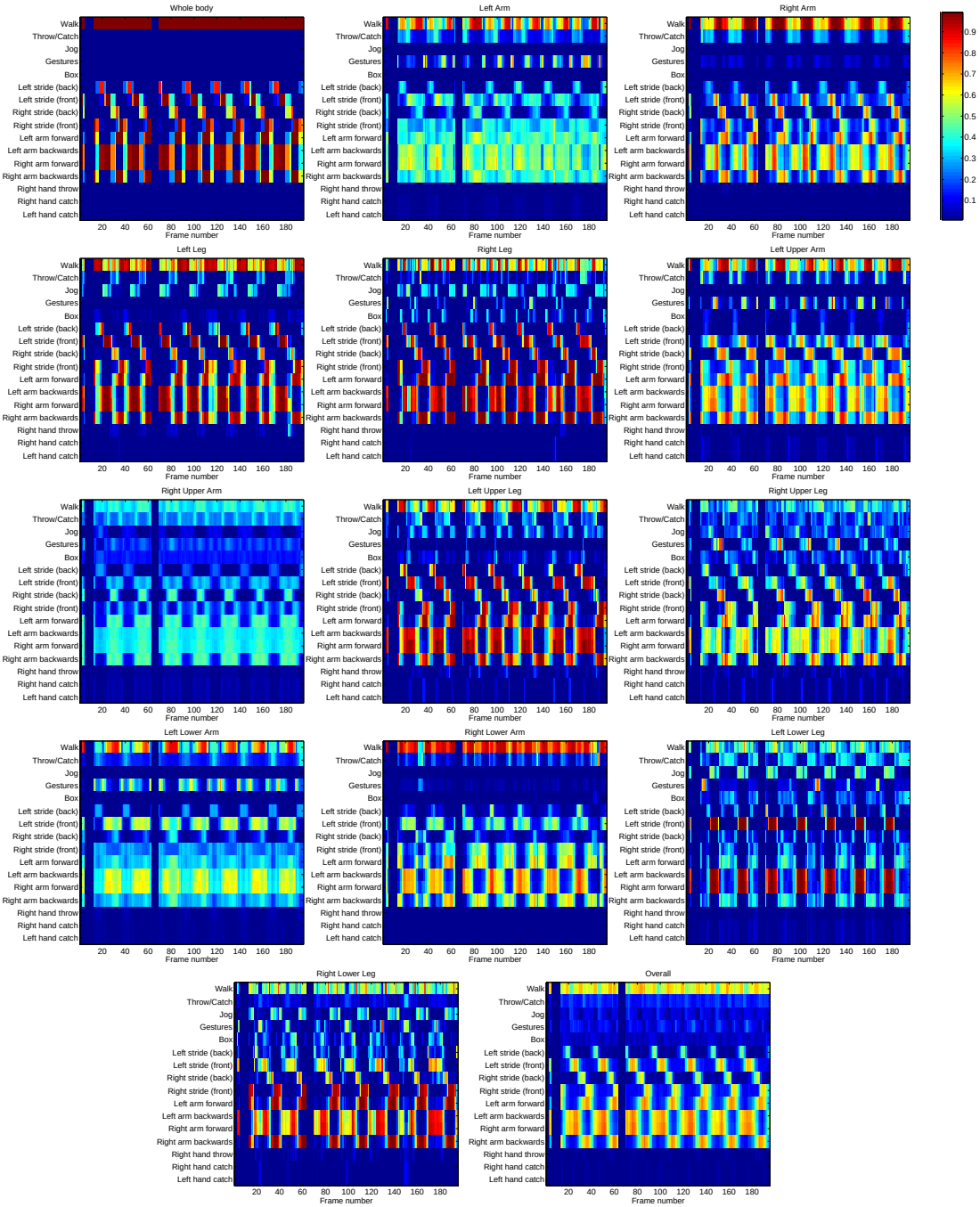



Figure 4.24: Recognition for the MCMs set on the *S1 Walking 1* sequence. For all models $n_{\mathcal{C}} = 100$ and $l_m = 3$. The probability of labels (vertical) for each frame (on horizontal) is colour coded. Zero probability is shown for the first 2 frames, frames 6–12 and 64–69 with missing pose information in the $l_m = 3$ long movement.

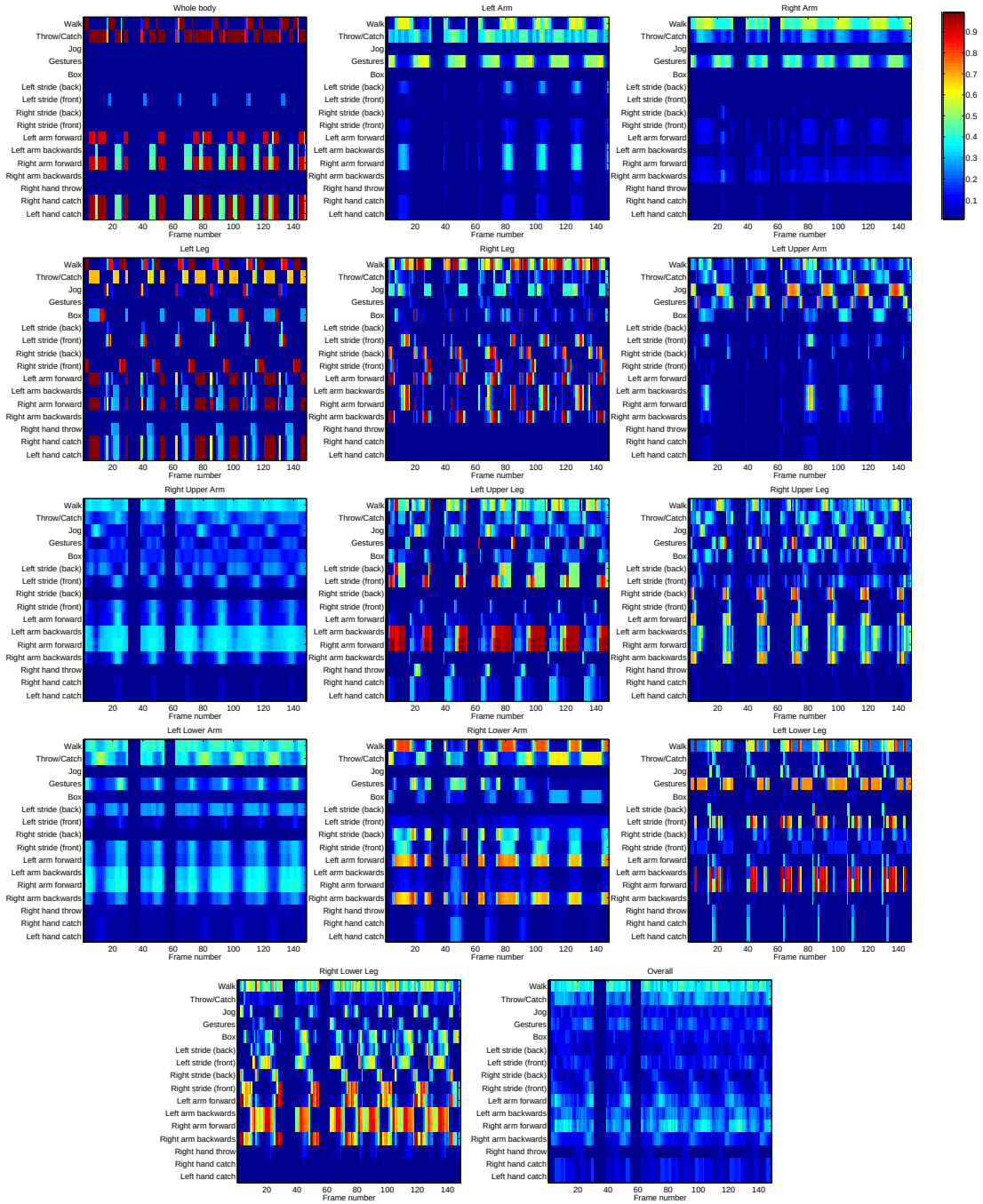



Figure 4.25: Recognition for the MCMs set on the *S3 Walking 1* sequence. For all models $n_{\mathcal{C}} = 100$ and $l_m = 3$. Zero probability is shown for the first 2 frames, frames 32–38 and 56–61 with missing pose information in the $l_m = 3$ long movement.

When recognition is poor, accuracy is far from the maximum one value, even bellow the random recognition. This is caused by the independence of labels, which may coexists or all have zero probabilities.

### 4.6.6 Recognition sensitivity

Action analysis follows visual tracking or other methods that at the current state of the art do not recover reliably the articulated configuration. It has missing body parameters (*e.g.* on extreme, providing blob tracking only), and other errors are caused by weak image input (*e.g.* due to occlusion and self-occlusions) or misalignment in the tracked model and real object. Hence the input data of the analysis is error prone.

The MCM model was tested against white noise with $0 \leq \sigma \leq 2$ variance, increasing in steps of 0.2, added to ground truth (*i.e.* MOCAP) model parameters of the S1 and S2 subjects from the HumanEva-I dataset *validate* partition. Evaluated with the HumanEva metric, equation (2.16), this added noise results in absolute and relative pose errors of between 26 and 280 mm, related to $\sigma$ as shown in figure 4.26. For reference, the optimised tracking errors reported in the next chapter are generally around 100mm, similar to those generated with $\sigma = 0.8$.

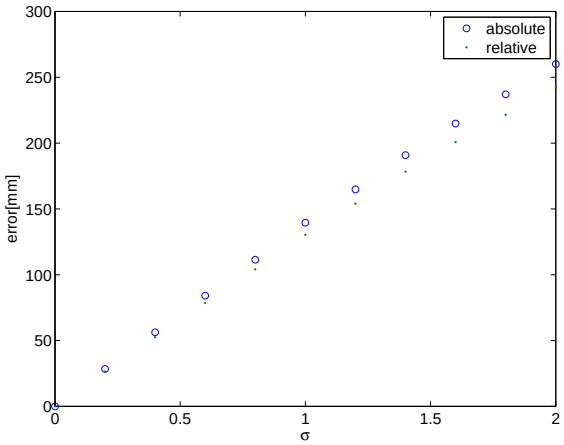



Figure 4.26: The absolute and relative errors in millimetres, generated by the Gaussian white noise of variance $\sigma$ of the normalised pose parameters.

The confusion matrices in figure 4.27 show how recognition performance degrades with $\sigma$, and a shift towards *Box* and *Throw/Catch* actions that are similar to other activities or have movements in common with them. This effect of noise will be remarked in chapter 6, while analysing the tracked parameters. It points out the necessity of accurate model

recovery for good action recognition.

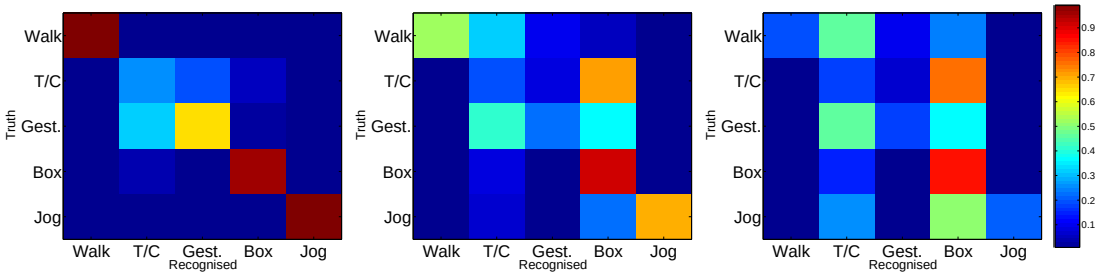



Figure 4.27: Action recognition with added noise. From left to right, confusion matrices for $\sigma \in \{0, 1, 2\}$ for $n_{\mathcal{C}} = 100$ and $l_m = 5$ MCM.

The accuracy is a function of the number of clusters and movement length, and depends on the added noise:

$$\zeta = \mathtt{F}(n_{\mathcal{C}}, l_m, \sigma). \tag{4.23}$$

For $n_{\mathcal{C}}$, $l_m$ and $\sigma$, the computed accuracies on the whole MCM, $\mathcal{M}_1$, from table 4.6, are represented visually in figure 4.28(a). Figure 4.28(b) represents the accuracy variation on the right leg MCM, $\mathcal{M}_6$. Both confirm the decrease in accuracy with increased noise. Similar to the results from section 4.6.3, more MCs strongly affect recognition, especially for long movements. Figures show that error tolerance is best is for $l_m = 15$ and $n_{\mathcal{C}} = 60$ or $n_{\mathcal{C}} = 80$, accuracy being kept high for increased $\sigma$.

While accuracy is higher for $l_m = 1$ the tolerance increases with $l_m$ up to $l_m = 15$. Optimal values for number of clusters are around $n_{\mathcal{C}} = 60$ and $n_{\mathcal{C}} = 80$.

On the other hand, figure 4.28(b) suggests that with a model with short BFV, the increase of either $n_{\mathcal{C}}$ or $l_m$ improves the recognition for increased noise. Although the accuracy is lower, since independent limbs are less descriptive than the whole pose for global action detection, the drop with $\sigma = 2$ is smaller for $\mathcal{M}_6$, around 50% compared to 10%–0% of the error-free ($\sigma = 0$) accuracy with $\mathcal{M}_1$.

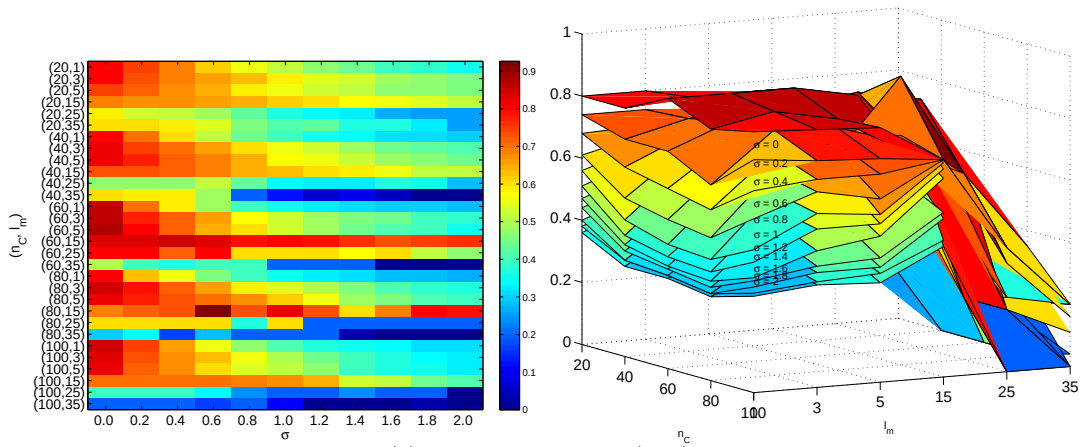
Since MCM formation uses K-means with random initialisation, repeated model construction will result different MCMs. Therefore their performance may also differ. Here only an arbitrary chosen MCM was considered.

## 4.7 Activity reasoning

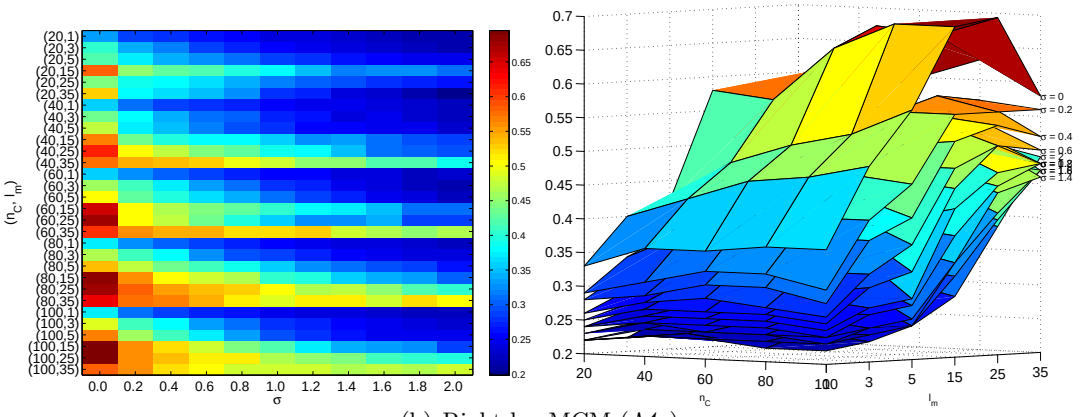
Generally, activities are composed of multiple actions. The MCM was used to identify actions such as *Walk*, *Throw/Catch*, *Gesture*, *etc*. These were mis-detected, since the

| $n_{\mathcal{C}}$ | $l_m$ | $\sigma$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 | 1.6 | 1.8 | 2.0 |
| 20 | 1 | 0.80 | 0.74 | 0.68 | 0.61 | 0.56 | 0.51 | 0.47 | 0.44 | 0.40 | 0.38 | 0.36 |
| 20 | 3 | 0.80 | 0.74 | 0.69 | 0.66 | 0.63 | 0.59 | 0.55 | 0.52 | 0.48 | 0.47 | 0.45 |
| 20 | 5 | 0.74 | 0.72 | 0.67 | 0.64 | 0.59 | 0.56 | 0.54 | 0.51 | 0.47 | 0.48 | 0.45 |
| 20 | 15 | 0.68 | 0.68 | 0.67 | 0.66 | 0.65 | 0.62 | 0.59 | 0.57 | 0.56 | 0.54 | 0.52 |
| 20 | 25 | 0.59 | 0.53 | 0.51 | 0.49 | 0.42 | 0.37 | 0.34 | 0.32 | 0.27 | 0.25 | 0.25 |
| 20 | 35 | 0.61 | 0.60 | 0.58 | 0.54 | 0.45 | 0.42 | 0.42 | 0.37 | 0.38 | 0.32 | 0.26 |
| 40 | 1 | 0.80 | 0.70 | 0.61 | 0.51 | 0.44 | 0.39 | 0.36 | 0.34 | 0.31 | 0.31 | 0.29 |
| 40 | 3 | 0.82 | 0.75 | 0.70 | 0.66 | 0.61 | 0.56 | 0.52 | 0.48 | 0.45 | 0.42 | 0.41 |
| 40 | 5 | 0.82 | 0.77 | 0.72 | 0.68 | 0.64 | 0.58 | 0.55 | 0.52 | 0.49 | 0.48 | 0.44 |
| 40 | 15 | 0.74 | 0.73 | 0.72 | 0.69 | 0.65 | 0.64 | 0.60 | 0.59 | 0.55 | 0.51 | 0.52 |
| 40 | 25 | 0.48 | 0.47 | 0.47 | 0.51 | 0.43 | 0.34 | 0.32 | 0.33 | 0.37 | 0.35 | 0.29 |
| 40 | 35 | 0.60 | 0.59 | 0.59 | 0.48 | 0.19 | 0.13 | 0.07 | 0.10 | 0.05 | 0.00 | 0.00 |
| 60 | 1 | 0.87 | 0.70 | 0.58 | 0.48 | 0.40 | 0.36 | 0.33 | 0.32 | 0.30 | 0.30 | 0.29 |
| 60 | 3 | 0.87 | 0.78 | 0.72 | 0.65 | 0.58 | 0.53 | 0.49 | 0.45 | 0.43 | 0.39 | 0.38 |
| 60 | 5 | 0.87 | 0.80 | 0.72 | 0.65 | 0.59 | 0.54 | 0.49 | 0.44 | 0.41 | 0.39 | 0.35 |
| 60 | 15 | 0.83 | 0.84 | 0.84 | 0.83 | 0.79 | 0.81 | 0.79 | 0.76 | 0.73 | 0.74 | 0.75 |
| 60 | 25 | 0.80 | 0.80 | 0.72 | 0.80 | 0.60 | 0.60 | 0.60 | 0.57 | 0.60 | 0.52 | 0.47 |
| 60 | 35 | 0.49 | 0.40 | 0.37 | 0.40 | 0.40 | 0.20 | 0.20 | 0.20 | 0.00 | 0.00 | 0.00 |
| 80 | 1 | 0.81 | 0.63 | 0.55 | 0.46 | 0.39 | 0.35 | 0.32 | 0.30 | 0.28 | 0.28 | 0.27 |
| 80 | 3 | 0.85 | 0.79 | 0.72 | 0.66 | 0.59 | 0.54 | 0.50 | 0.45 | 0.41 | 0.38 | 0.37 |
| 80 | 5 | 0.81 | 0.78 | 0.73 | 0.70 | 0.67 | 0.62 | 0.56 | 0.49 | 0.46 | 0.41 | 0.40 |
| 80 | 15 | 0.69 | 0.72 | 0.74 | 0.93 | 0.74 | 0.82 | 0.74 | 0.60 | 0.68 | 0.80 | 0.79 |
| 80 | 25 | 0.60 | 0.60 | 0.60 | 0.60 | 0.37 | 0.60 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 |
| 80 | 35 | 0.29 | 0.35 | 0.17 | 0.29 | 0.20 | 0.20 | 0.20 | 0.04 | 0.00 | 0.00 | 0.00 |
| 100 | 1 | 0.84 | 0.75 | 0.65 | 0.56 | 0.46 | 0.42 | 0.37 | 0.36 | 0.34 | 0.32 | 0.31 |
| 100 | 3 | 0.83 | 0.73 | 0.67 | 0.63 | 0.56 | 0.51 | 0.44 | 0.39 | 0.37 | 0.34 | 0.33 |
| 100 | 5 | 0.82 | 0.72 | 0.67 | 0.59 | 0.54 | 0.51 | 0.47 | 0.40 | 0.37 | 0.35 | 0.32 |
| 100 | 15 | 0.70 | 0.70 | 0.70 | 0.70 | 0.69 | 0.67 | 0.64 | 0.53 | 0.51 | 0.44 | 0.42 |
| 100 | 25 | 0.40 | 0.40 | 0.40 | 0.35 | 0.24 | 0.20 | 0.20 | 0.24 | 0.20 | 0.20 | 0.00 |
| 100 | 35 | 0.20 | 0.20 | 0.20 | 0.16 | 0.20 | 0.11 | 0.00 | 0.00 | 0.00 | 0.04 | 0.00 |

Table 4.6: Accuracy $\zeta$ variation for added noise $\sigma \in \{0, 0.2 \dots 2\}$. The accuracy decreases with added noise. Optimal values for number of clusters are $n_{\mathcal{C}} = 60$ and $n_{\mathcal{C}} = 80$, with the error tolerance being the best for $l_m = 15$ and $n_{\mathcal{C}} = 60$ or $n_{\mathcal{C}} = 80$ with accuracy being kept high for increased $\sigma$.




(a) Whole pose MCM ($\mathcal{M}_1$)




(b) Right leg MCM ($\mathcal{M}_6$)

Figure 4.28: Accuracy $\zeta$ variation for added noise $\sigma \in \{0.0, 0.2 \ldots 2.0\}$. With increasing noise, the recognition is degrading.

movements reflect only the pose sequence they cover, while activities have longer duration. To enhance the detection of activities as well as actions, the first option is to increase the length $l_m$ of the movement. However, $l_m > 15$ does not improve activity recognition, and also affects low level action detection (sections 4.6.3 and 4.6.6).

The combination of low level labels at a higher level of analysis is an alternative. For this, the simplest inference is the expectation of the activity labels over several, $N_{activity} \gg l_m$, frames and, at an extreme, over the whole sequence. The confusion matrices resulting from classifying each complete sequence (*i.e.* not each movement individually) are shown in figure 4.29.

The stationary activities are misclassified as *Gesture*, because of the similarities of the long, standing poses. Detecting over short periods, Mather *et al.* [10] shows, is biologically more plausible, Therefore in future, more diverse short actions should be detected and assembled in activities with composition rules (fixed or learnt) *e.g.* with a Stochastic Context Free Grammar [58] as used by Ivanov [51].

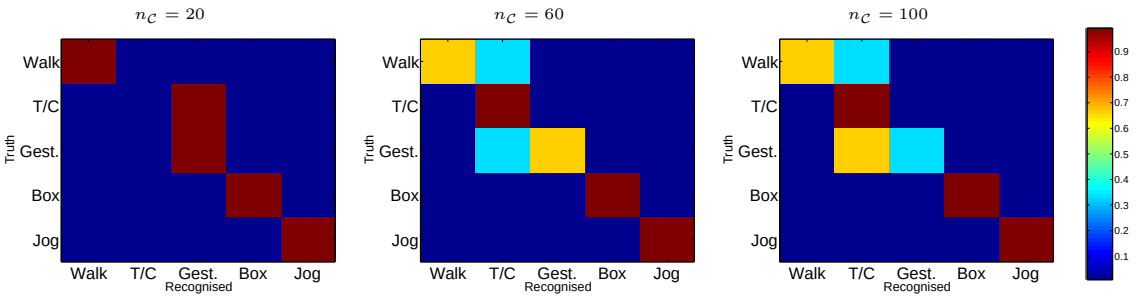



Figure 4.29: Confusion matrices of sequence classification. Each HumanEva sequence is classified into the activity which has the most corresponding action.

## 4.8 Summary

This chapter discusses articulated motion prediction and behaviour modelling for action recognition. Models were trained and verified with the ground truth provided with the HumanEva dataset.

First, to predict the motion from the previous one or more poses, three different models, all with a learning algorithm and associated procedures to compress and represent the poses, were developed. The dynamic models were learnt unsupervised from the HumanEva dataset, and were tested visually to assess whether the generated synthetic poses resembled non-intentional, but realistic motion.

The PTM model has no memory, but is a HMM with discrete, learnt, pose cluster states, trained unsupervised. The generated motion consists of discrete poses, thus it is discursive, especially for a low number of PCs. The model has the fault that it uses the current pose only for the prediction.

The CTPTM extends the PTM, with continuous transitions, by means of learning the transition durations between states, together with their probabilities. Therefore, the generated motion is smooth and visually resembles a human motion, and is a mix of sort sequences of the learnt activities. CTPTM allows implicit time modelling that adapts to changing frame rate, and to slower or faster movements. However, this causes incompatibility with a partitioned tracking.

The MCM overcomes the discrete pose representation and the single pose based estimation, *i.e.* the $1^{st}$ order Markovian requirement, by replacing poses by movements and PC states by MC states. Furthermore, traditional transitional probabilities of a HMM are replaced by continuous Gaussian transitions, which estimate next BFVs and indirectly define the next MC state.

To find similar poses and movements, *i.e.* to cluster them, EM was employed. Improved clustering methods possibly would generate more representative pose and movement clusters, however their evaluation was not the subject of this modelling.

In the second part of the chapter, the MCM model is extended for behavioural analysis. For each MC, the probabilities of action labels are learnt by supervised learning. The recognition performance tests show that the MCM achieves good recognition rates for the general activity, and the periodic aspects of the repetitive actions are also well detected. MC uniformity test suggests that longer movements and more clusters result in more individualised MCs, however it is interesting to note that increasing the length of a movement, or the number of movement clusters, has adverse effects. For low level, detailed actions, movement length should be short and modelled with many clusters, while for global actions longer movements with limited numbers of clusters are preferred. Further, it was noted that limb based MCM also recognise full body and other limb defined actions, suggesting the expected high parameter inter-dependence.

The advantages of the developed MCM are:

- dual applicability for prediction and recognition;
- prediction of both periodic and aperiodic actions;

- separation of MCM training into an unsupervised and a supervised phase, therefore new action labels can be added, replaced or removed, incrementally, independent of the MC training and the dynamic model;
- the detected and recognised feature set is arbitrary. This applies to either the whole parameter set, or a subset;
- the MCM set defines a pool of classifiers available for higher level analysis;
- both actions and simple activities with a medium duration are recognised.

The next chapter applies these models for motion prediction in visual human tracking, and will argue whether or not they are appropriate in a particle filter framework.

# Chapter 5

# Articulated human tracking

The requirements and interests for articulated tracking were explained in the previous chapters. Therefore, this chapter focuses on tracking that recovers together with the 3D position also the pose of the human body. The tracking methods described aim to deal with scenes of medium complexity, without small details of the subject, but going beyond blob tracking to which much other behavioural research is limited. The tracking problem is complex, because the model is high dimensional, and observations are noisy, with self occlusions, occlusions and environmental changes.

Introduced in section 2.2.2, *Particle filters* (PF) are stochastic filters that effectively track high dimensional and non-Gaussian models. For this reason, they are used frequently for human tracking (section 2.2.2, table 2.5).

For clarity, first the evaluation methodology and the notations for stochastic distribution and for random sampling are defined. Then, starting from Kalman Filters, the basic PF with SIR are reviewed and evaluated for tracking the Articulated Hierarchical Human Model from chapter 3. Similarly, both the Partitioned and the Annealed Particle Filters are recalled. Then, their generalisation, the *Hierarchical Partitioned Particle Filter* is introduced and specialised for tracking the Articulated Hierarchical Human Model, and further compared to previous articulated tracking algorithms. Next, the dynamic model is implemented with the *Movement Cluster Model* (MCM) from chapter 4.

As for most tracking algorithms, parameters of the HPPF-MCM require an empirical adjustment. Therefore, structured analysis and tuning is performed, while other improvements for likelihoods, priors, estimates and particle distribution are also proposed.

Finally, after the effect of reducing the number of cameras is analysed, tracking results are objectively evaluated with the HumanEva I and II datasets, and subjectively with the CAVIAR and the i-LIDS datasets.

## 5.1 Evaluation methodology

For this chapter, the HumanEva datasets (section 2.6.1) provide the main evaluation data. First, it allows visual interpretation of the tracking with the recovered poses either projected and superimposed the original images or rendered in 3D space. For quantitative evaluation, in addition to inspection, algorithms are compared with the metrics from section 2.7, either plotting the error for a test sequence or, with equation (2.16), the mean error (absolute, equation (2.14), and relative, equation (2.15)) averaged for the seven test sequences from table 5.1. These include both moving and static body position and the sequences with labelled actions from chapter 4. However, the test set design is limited by the missing *Throw/Catch* ground truth for subjects S1 and S3, and by the processing time of many sequences run with all experiments performed later in the chapter.

Both 2D and 3D errors are presented for completeness, where the space allows, however these generally do not provide additional information relative to the 3D absolute error.

| **No.** | **Name** | **Frame range** | **Length [s]** |
|---|---|---|---|
| 1 | *S1 Walking 1* | 6–588 | 9.75 |
| 2 | *S1 Jog 1* | 6–366 | 6.05 |
| 3 | *S1 Gesture 1* | 6–393 | 6.50 |
| 4 | *S3 Walking 1* | 6–447 | 7.40 |
| 5 | *S3 Jog 1* | 6–399 | 6.60 |
| 6 | *S3 Gesture 1* | 6–531 | 8.80 |
| 7 | *S2 Throw/Catch 1* | 6–549 | 9.10 |

Table 5.1: Test sequences for articulated human tracking from the validation partition of the HumanEva dataset. Sequences are down-sampled from 60fps to 20fps and include every third frame of the whole validation partition.

For compatibility with lower frame rate videos, HumanEva sequences are down-sampled by 1/3 to 20fps, comparable with 25fps CAVIAR and i-LIDS datasets.

The optimised tracking is also tested with CAVIAR and i-LIDS sequences, however only by visual inspection.

## 5.2 Stochastic tracking

### 5.2.1 Uniform and normal distributions. Sampling

The uniform and the normal are the most important distributions employed generally in stochastic methods and in this thesis. Frameworks such as Monte Carlo Markov Chain, Unscented Kalman Filters and PFs represent and manipulate the tracked distributions implicitly with a finite number of samples drawn from the distribution. For clarity, the following notation is adopted with the corresponding sample generation implemented by the tracking algorithm.

The most common distribution is uniform distribution over a continuous range $[a, b)$.

**Notation.** *$\mathcal{U}(x; a, b)$ is the continuous uniform density of $x$ with values in the range $[a, b)$.*

$$\mathcal{U}(x; a, b) = \begin{cases} 0 & x < a \\ \frac{1}{b-a} & a \leq x < b \\ 0 & x \geq b \end{cases} \tag{5.1}$$

.

To sample a value $x \sim \mathcal{U}(x; a, b)$ from this distribution, the uniform random number generator is available inbuilt in most of the high level programming languages.

If such a function is not provided then [222, pp.275–286] supplies algorithms for uniform sample generation.

**Notation.** *$\mathcal{U}\{x; (p_k, \xi_k)\}$ is the discrete uniform distribution of $x$ with probabilities $p_k = \mathcal{P}(x = \xi_k)$ for the $k$ possible values.*

Sampling one value $x$ appeals to the sampling for the uniform continuous distribution:

$$\varsigma \sim \mathcal{U}(s; 0, 1), \tag{5.2}$$

and selects $x = \xi_s$ where

$$\sum_{j=1}^{s} p_j \leq \varsigma < \sum_{j=1}^{s+1} p_j. \tag{5.3}$$

For a vector x with independent elements, each $x_k$ is generated separately, as described above.

**Notation.** *$\mathcal{N}(\mathrm{x};\mathrm{m},\mathbf{P})$ is the Gaussian (or normal) p.d.f. of a d-element vector* x*, with mean* m *and covariance* $\mathbf{P}$*:*

$$\mathcal{N}(\mathrm{x};\mathrm{m},\mathbf{P}) = (2\pi)^{-\frac{d}{2}}|\mathbf{P}|^{-\frac{1}{2}}e^{-0.5(\mathrm{x}-\mathrm{m})^T\mathbf{P}^{-1}(\mathrm{x}-\mathrm{m})}. \tag{5.4}$$

For the uni-variate case, with $\mathrm{x} = x$ scalar, if no in-built function such as the Matlab `randn` exists, then the Box-Muller transform generates the Gaussian samples $x \sim \mathcal{N}(x;0,1)$ using two independent uniform $y_1, y_2 \sim \mathcal{U}(y_i;0,1)$ distributions [222, p.289]:

$$x = \sqrt{-2\ln y_1}\cos 2\pi y_2. \tag{5.5}$$

For a multivariate vector x, drawing from $\mathcal{N}(\mathrm{x};\mathrm{m},\mathbf{P})$ results in [223, p.368]:

$$\mathrm{x} = \mathrm{m} + \sqrt{\mathbf{P}}\mathrm{y}, \tag{5.6}$$

where y is a vector of independent and Gaussian-distributed zero mean and unit standard deviation, $y_i \sim \mathcal{N}(y;0,1)$ values.

### 5.2.2 Kalman filters

A *Kalman filter* (KF) uses the dynamic system model

$$\mathrm{x}_k = \mathtt{f}_k(\mathrm{x}_{k-1}, \mathrm{v}_{k-1}) \tag{5.7}$$

and computes the system state vector $\mathrm{x}_k$ based on the noisy observations $\mathrm{y}_k$. The system dynamic is expressed by the known function $\mathtt{f}_k$. The state $\mathrm{x}_k$ is hidden and can be observed by a $\mathrm{z}_k$ measurement vector through the known $\mathtt{h}_k$ observation model:

$$\mathrm{z}_k = \mathtt{h}_k(\mathrm{x}_k, \mathrm{w}_k) \tag{5.8}$$

The conventional Kalman filter was designed to work with linear system and observation models:

$$\mathrm{x}_k = \mathrm{F}_{k-1}\mathrm{x}_{k-1} + \mathrm{v}_{k-1}, \tag{5.9}$$

$$\mathrm{z}_k = \mathrm{H}_k\mathrm{x}_k + \mathrm{w}_k, \tag{5.10}$$

where the normally distributed $\mathrm{v}_k$ system and $\mathrm{w}_k$ measurement errors are

$$\mathrm{v}_k \sim \mathcal{N}(\mathrm{w}_k; 0, \mathbf{Q}_k), \text{ and} \tag{5.11}$$

$$\mathrm{w}_k \sim \mathcal{N}(\mathrm{v}_k; 0, \mathbf{R}_k), \tag{5.12}$$

with $\mathbf{Q}_k$ and $\mathbf{R}_k$ covariances.

The KF consists of two steps. The first estimates the current state and the variance, using observations up to the current state:

$$\mathrm{x}_k^- = \mathrm{F}_{k-1}\mathrm{x}_k^+, \tag{5.13}$$

$$\mathbf{P}_k^- = \mathrm{F}_{k-1}\mathbf{P}_k^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{5.14}$$

Next, the current observation is integrated into the model:

$$\mathrm{x}_k^+ = \mathrm{x}_k^- + \mathrm{K}_k\left(z_k - \mathbf{H}_k x_k^-\right), \tag{5.15}$$

$$\mathrm{P}_k^+ = \left(\mathrm{I} - \mathrm{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-, \tag{5.16}$$

$$\mathrm{K}_k = \mathbf{P}_k^-\mathrm{H}_k^T\left(\mathrm{H}_k\mathbf{P}_k^-\mathrm{H}_k^T + \mathbf{R}_k\right)^{-1}. \tag{5.17}$$

The linear constraints, equations (5.9) and (5.10), are strong limitations for such a system. The *Extended Kalman filter* (EKF) for non-linear systems with additive noise

$$\mathrm{x}_k = \mathtt{f}_k(\mathrm{x}_{k-1}) + \mathrm{v}_{k-1}, \tag{5.18}$$

$$z_k = \mathtt{h}_k(\mathrm{x}_k) + \mathrm{w}_k, \tag{5.19}$$

assumes $\mathtt{f}_k$ and $\mathtt{h}_k$ linear in small intervals. EKF tracking is similar to the KF equations (5.9) to (5.17), where $\mathrm{F}_k$ and $\mathrm{H}_k$ are replaced with locally linearised $\mathrm{F}_k^*$ and $\mathrm{H}_k^*$:

$$\mathrm{F}_{k-1}^* = \left.\frac{\partial f_{k-1}}{\partial x_{k-1}}\right|_{x_{k-1}=x_{k-1}^+} \text{ and} \tag{5.20}$$

$$\mathrm{H}_k^* = \left.\frac{\partial h_k}{\partial x_k}\right|_{x_k=x_k^-}. \tag{5.21}$$

EKF is computationally expensive, since in each iteration step it requires the Jacobians

defined by equations (5.20) and (5.21); it works with non-linearity up to the first order and with normally distributed noise.

In the *Unscented Kalman filter* (UKF), sample points that capture the Gaussian parameter distribution are propagated through the non-linear system, generating the posterior distribution of the variables. Unscented transformation obtains the

$$\mathrm{y}(i) = \mathtt{f}(\mathrm{x}(i)), \tag{5.22}$$

by establishing deterministically $2n+1$ weighted sample (or sigma) points $\mathrm{S}_i = \{w(i), \mathrm{x}(i)\}$ which completely capture the true mean and covariance of X.

Then, the mean and covariance of the posterior is

$$\overline{\mathrm{y}} = \sum_{i=0}^{2n} w(i)\mathrm{y}(i) \quad \text{and,} \tag{5.23}$$

$$\mathbf{P}_y = \sum_{i=0}^{2n} w(i)\left(\mathrm{y}(i) - \overline{\mathrm{y}}\right)\left(\mathrm{y}(i) - \overline{\mathrm{y}}\right)^T. \tag{5.24}$$

The UKF is the straightforward application [103] of the unscented transformation. It has an initialisation step that computes the mean and covariance of the initial distribution and builds the parameter vector including the system parameters, the process and the measurement noises. For every measurement, the sigma points of the current distribution are computed; the sigma points are propagated through the system equations using the unscented transformation, equation (5.22). This, equivalent to the time update phase of the KF, results in the a priori sigma points of parameters, $\mathrm{x}_k^-$, and the predicted observations, $\mathrm{z}_k^-$. Further, equation (5.24) provides the a priori parameter covariance $\mathrm{P}_k^-$. Then, the measurement update phase computes the posterior prediction and covariance in the same way as in equations (5.15)–(5.16). Although UKF works with non linear systems up to the second order, it is limited to Gaussian models.

### 5.2.3 Particle Filter basics and Sequential Importance Resampling

The basic PF is the Sequential Importance Resampling (SIR) algorithm [60], shown in algorithm 8. The parameter distribution at time $t$ is represented by the set of $n_p$ parameter vectors, the particles $\mathrm{p}_t(i)$, $i = 1..n_p$. In each SIR iteration, the new distribution $\Phi_t = \{\mathrm{p}_t(i)\}_{i=1}^{n_p}$ is estimated from the previous distribution $\Phi_{t-1}$, by incorporating the

**Algorithm 8**: Sequential Importance Resampling (SIR) (based on [60])

**Input**: $\Psi_{t-1} = \{\mathrm{p}_{t-1}(i)\}_{i=1}^{n_p}$ – previous particle set and
$\mathrm{O}_t$ – current observation
**Output**: $\Psi_t$ – current particle set

1 **for** $i = 1 : n_p$ **do**
2 $\quad \mathrm{p}_t(i) \sim q(\mathrm{p}_t|\mathrm{p}_{t-1}(i))$ `// draw new samples`
3 $\quad \tilde{w}_t(i) = \lambda(\mathrm{O}_t|\mathrm{p}_t(i))$ `// evaluate the importance weights up to normalising constant`
4 **end**
5 $t = \sum_{i=1}^{n_p} \tilde{w}_t(i)$ `// total weight`
6 **for** $i = 1 : n_p$ **do**
7 $\quad w_t(i) = \frac{\tilde{w}_t(i)}{t}$ `// normalise`
8 **end**
9 $\{\mathrm{p}_t(i), -\}_{i=1}^{n_p} =$ `Resample`$(\{\mathrm{p}_t(i), w_t(i)\}_{i=1}^{n_p})$ `// resample using Algorithm 9`
10 $\Psi_t = \{\mathrm{p}_t(i)\}_{i=1}^{n_p}$

current observations $\mathrm{O}_t$. First, the assumed motion model, generically represented by equation (5.7), generates a new distribution in line 2. Then, the observation likelihood is computed for all samples of the distribution, resulting in the un-normalised weights of particles. The propagation, lines 1–4, is similar to the distribution propagation with the unscented transform of the UKF.

The weights are normalised, lines 5–8, and finally the resampling step avoids the degeneration of the particle set. The SIR performs a resampling at each iteration, generating equally weighted particles, and therefore weights are not maintained through iterations.

Algorithm 9 [60] resamples uniformly the weighted particles, by means of their cumulative sum. The resampling does not alter the input distribution, only the particle set representing it. These initially have unequal weights while in the final set they are equalised.

The two key elements of the PF are the motion update, defined by distribution $q$, and the likelihood $\lambda(\mathrm{O}_t|\mathrm{p}_t(i))$, specific for every tracking problem.

### Human tracking with the SIR particle filter

To track the *Articulated Hierarchical Human Model* (AHHM) defined in chapter 3, the pose vector p is adopted as the particle $\mathrm{p}_t(i)$ of the SIR filter.

The simplest motion model is the *zero-order Gaussian motion* (0GM) (section 2.3.2) which assumes a change of parameters described by a Gaussian white noise. Therefore,

**Algorithm 9**: Resampling (based on [60])

**Input**: $\{\mathrm{p}_t(i), w_t(i)\}_{i=1}^{n_p}$
**Output**: $\{\hat{\mathrm{p}}_t(j)\}_{j=1}^{n_p}$

1 $c_1 = w_t(1)$ `// initialise cumulative sum of weights (CSW)`
2 **for** $i = 2 : n_p$ **do**
3 $\quad c_i = c_{i-1} + w_t(i)$ `// construct CSW`
4 **end**
5 $i = 1$ `// start sampling with the first particle`
6 $u_1 \sim \mathcal{U}(u; 0, {n_p}^{-1})$ `// draw stochastic starting point`
7 **for** $i = 1 : n_p$ **do**
8 $\quad u_j = u_1 + {n_p}^{-1}(j-1)$ `// next sample`
9 $\quad$ **while** $u_j > c_i$ **do**
10 $\quad\quad i = i + 1$
11 $\quad$ **end**
12 $\quad \hat{\mathrm{p}}_t(j) = \mathrm{p}_t(i)$ `// resample`
13 **end**

drawing from distribution $q$ in line 2 of algorithm 8 results in the particle update:

$$\mathrm{p}_t(i) = \mathrm{p}_{t-1}(i) + \mathrm{v}_0, \tag{5.25}$$

where

$$\mathrm{v}_0 \sim \mathcal{N}(\mathrm{v}; 0, \mathbf{P}) \tag{5.26}$$

is sampled from a Gaussian density with zero mean and $\mathbf{P}$ covariance.

With the global likelihood $\lambda_G(\mathrm{O}|\mathrm{p})$ from section 3.3.5, with $n_p = 600$ particles, the SIR filter provides the baseline evaluation for human tracking. The tracked frames 6–57 of the *S1 Walking 1* sequence are visualised with the first camera, C1, view (figure 5.1) and with the 3D reconstruction (figure 5.2).

The tracking recovers and maintains an estimate of the 3D position of the body over the whole sequence, however the recovered pose is poor. Although the trunk localisation is good, limbs are incorrectly tracked. In particular, the lower limb parameters float around the middle of the parameter range. Errors over the whole sequence for both position and pose (figures 5.3) also show high errors. Both remarks suggest that the likelihoods do not direct the particles and cannot optimise them with a low number of particles spread across a wide parameter range, problems in common with [117, 128]. This is intractable without a directed search in the parameter space.

Further, randomly sampled, independent limbs are unlikely to provide concurrently a

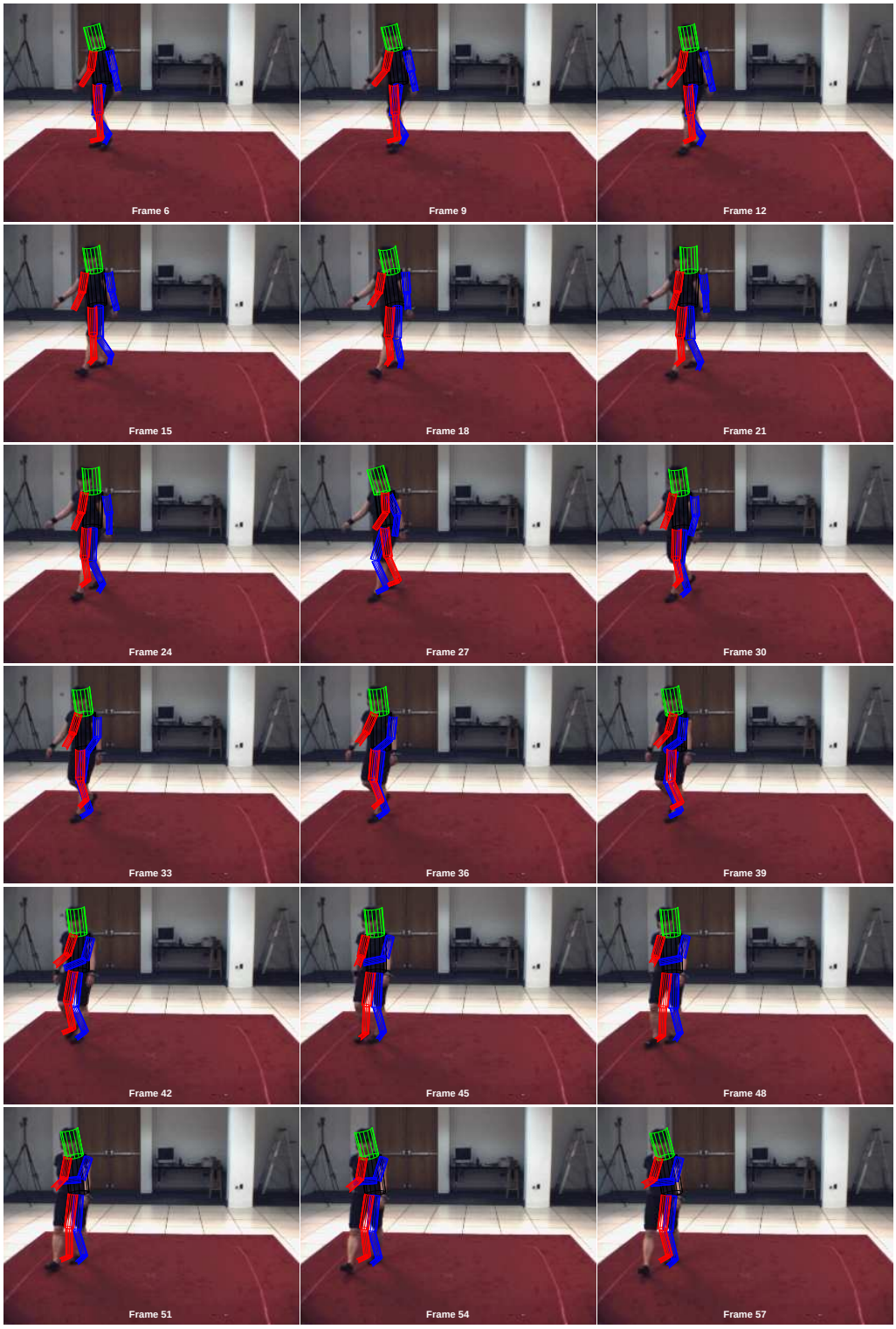



Figure 5.1: PF-SIR tracking. The recovered pose is superimposed with the camera C1 view [◊].

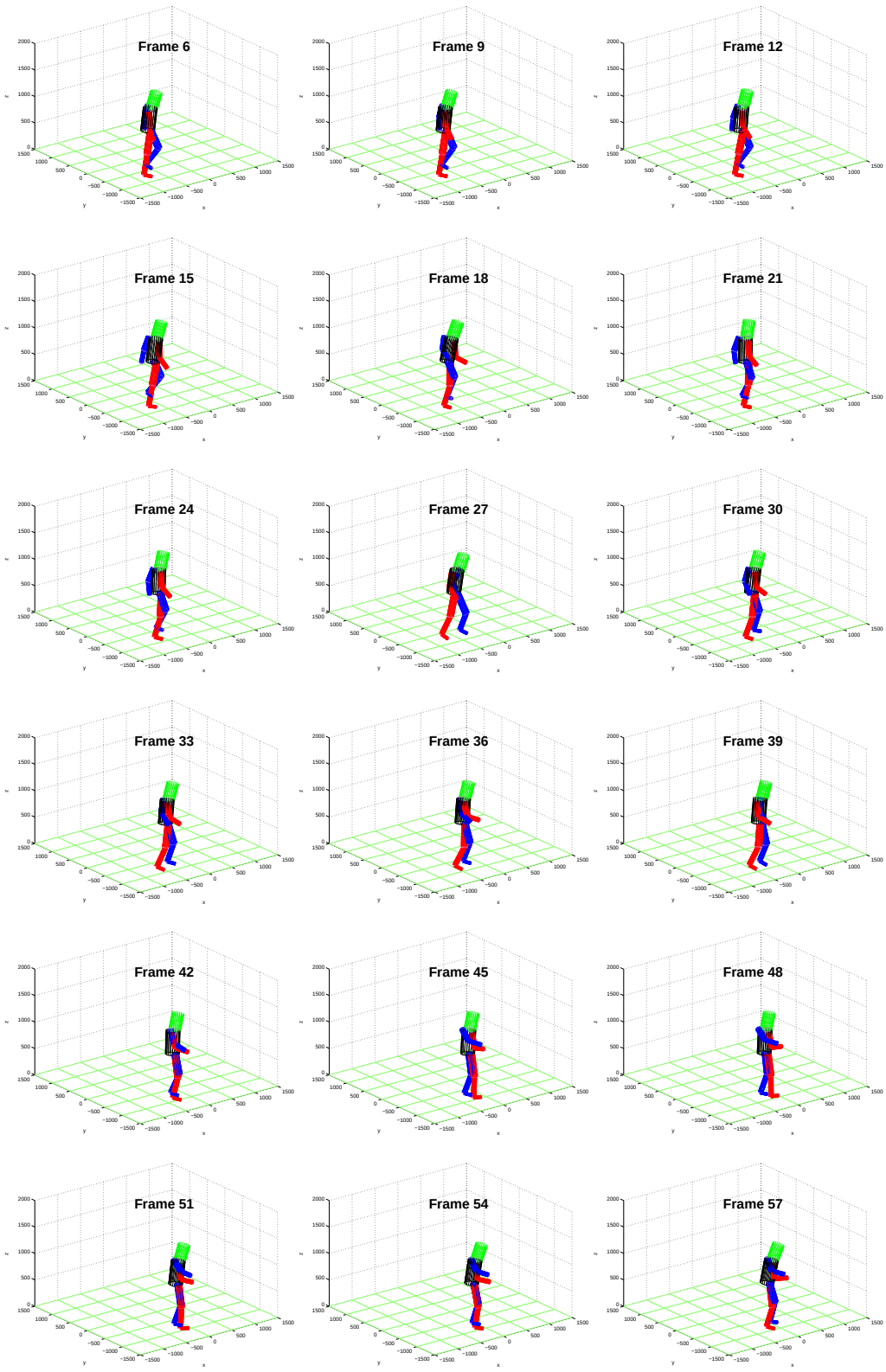



Figure 5.2: PF-SIR tracking. The 3D reconstruction of the *S1 Walking 1* sequence [◊].

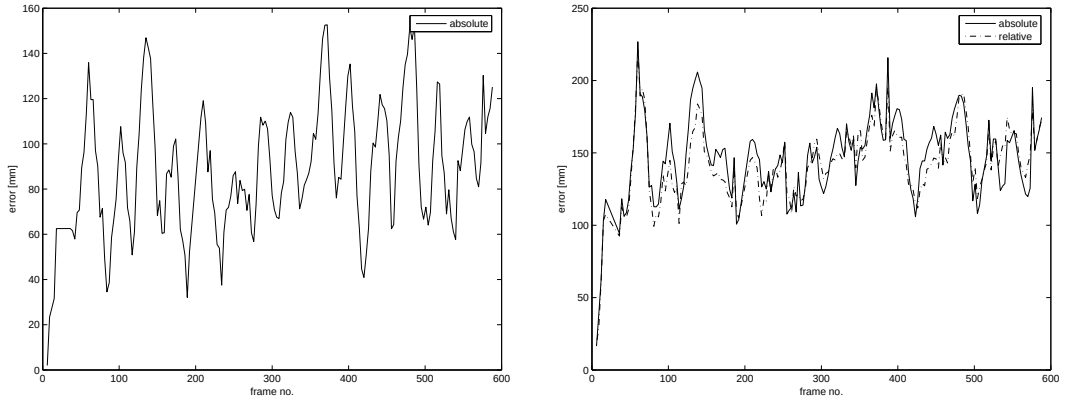



Figure 5.3: PF-SIR tracking: *S1 Walking 1* 3D error. The absolute body centre and the general position and pose (absolute and relative) errors suggest poor tracking over the whole sequence.

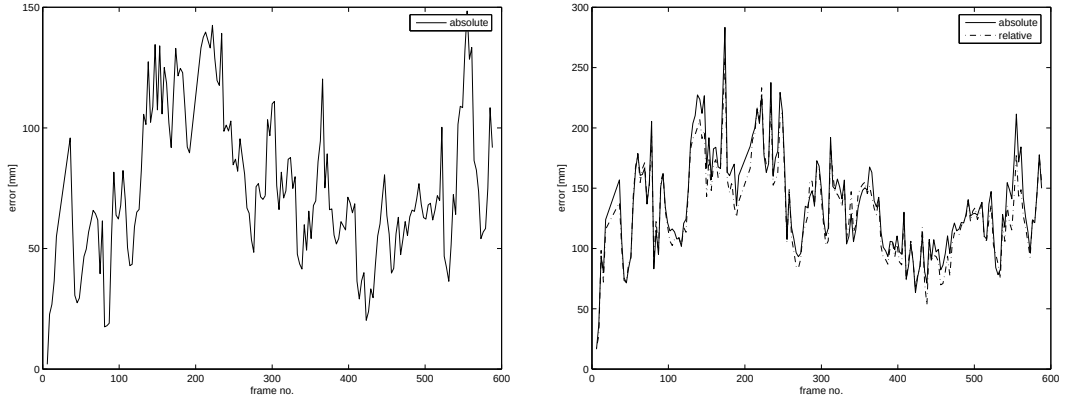



Figure 5.4: PF-SIR tracking with elimination: *S1 Walking 1* 3D error. Particles with low weights are removed and the other weights are emphasised. The absolute body centre, and the general position and pose errors (absolute and relative) are more stable and with lower peaks.

good fit. Therefore the global likelihood is not discriminatory enough to eliminate the lowest likelihood particles.

As a proposed initial PF enhancement, the particles are thresholded, and those with low-likelihoods are eliminated explicitly after the evaluation. Remaining particle weights are stretched to emphasise their differences.

This procedure results in reduced pose and position errors (figure 5.4), with 3D absolute mean errors 152.5 mm and respectively 141.7 mm (for numerical comparison see table 5.3); the trunk, the upper limbs and also one or two of the lower limbs are tracked. This low-likelihood elimination strategy will be analysed and evaluated in section 5.4.6.

Additional gain is expected from *clever* particle filtering that maximises concurrently all limb fitness, overcomes the local likelihood minima and generates a global maximum.

Sections 5.2.4 and 5.2.5 present briefly existing altered versions of the SIR particle filter, while section 5.3 will combine the advantages of the two by an explicitly hierarchical and partitioned particle filter. Then, to direct particles by the learnt human motion, the motion model described in section 4.5 is integrated with the HPPF.

### 5.2.4 The Partitioned Particle Filter

The Partitioned Particle Filters (PPF) of MacCormick and Isard (introduced in section 2.2.2) divide the parameter space into independent partitions. The method suggests high potentials for *independent* limb tracking. However, for AHHM, limbs rely on the defining higher level limb or on the global position. The PPF does not fit this dependence.

### 5.2.5 The Annealed Particle Filter

Another, high dimensional articulated model tracker from section 2.2.2, the Annealed Particle Filter (APF), overcomes local minima creating "heated" particles by replacing their image likelihood function $\lambda(\mathrm{O}|\mathrm{p})$ on the $m$-th annealing level with

$$\lambda_m(\mathrm{O}|\mathrm{p}) = [\lambda(\mathrm{O}|\mathrm{p})]^{\beta_m}, \tag{5.27}$$

with $\beta_0 > \beta_1 > \cdots > \beta_{n_L}$ annealing parameters selected to maintain the desired particle survival rate [117].

Further, in each time instance $t$, the proposed dynamic hierarchical tracking generates samples from distribution $q_m$ with covariance, $\mathbf{P}_m$, proportional to the covariance of the particle set $\{{}^m\mathrm{p}_t(i)\}$ on level $m$:

$$\mathbf{P}_m = c_{apf}\frac{1}{n_p}\sum_{i=1}^{n_p}[{}^m\mathrm{p}_t(i) - {}^m\overline{\mathrm{p}}_t]\cdot[{}^m\mathrm{p}_t(i) - {}^m\overline{\mathrm{p}}_t]^T, \tag{5.28}$$

where ${}^m\overline{\mathrm{p}}_t$ is the average particle.

The tracking shown in figures 5.5 and 5.6 uses $n_p = 60$ particles on each of the 10 layers (equivalent to the 600 particles of the PF-SIR), with the global edge and silhouette likelihood $\lambda_G(\mathrm{O}|\mathrm{p})$ from section 3.3.5. The covariance $\mathbf{P}_m$ is diagonal, and initialised at level $m = 1$ with the covariance of the training data and updated on each level with equation (5.28) and the covariance proportionality $c_{apf} = 0.3$.

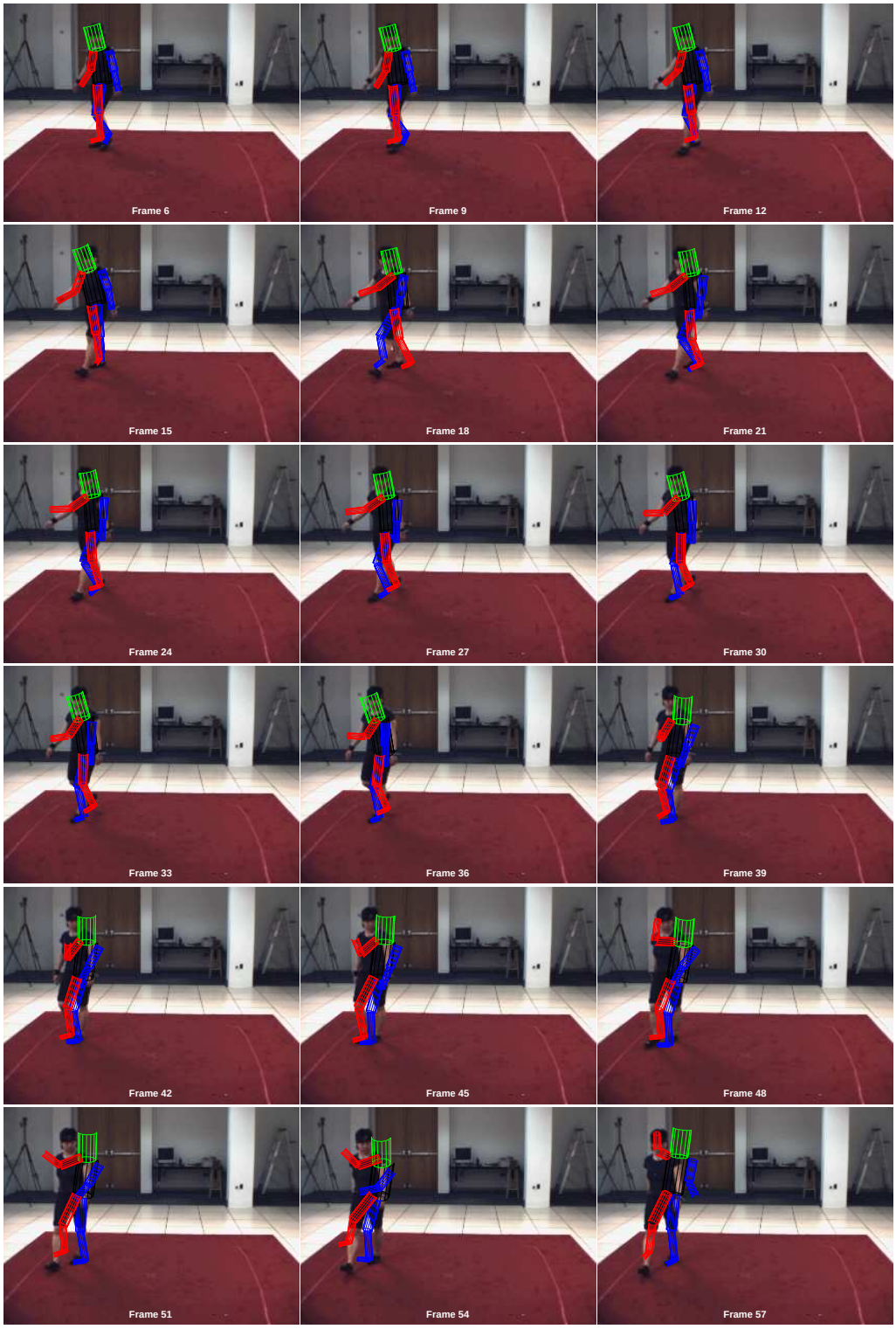



Figure 5.5: APF tracking. The recovered *S1 Walking 1* poses are superimposed with the camera C1 view [◊].

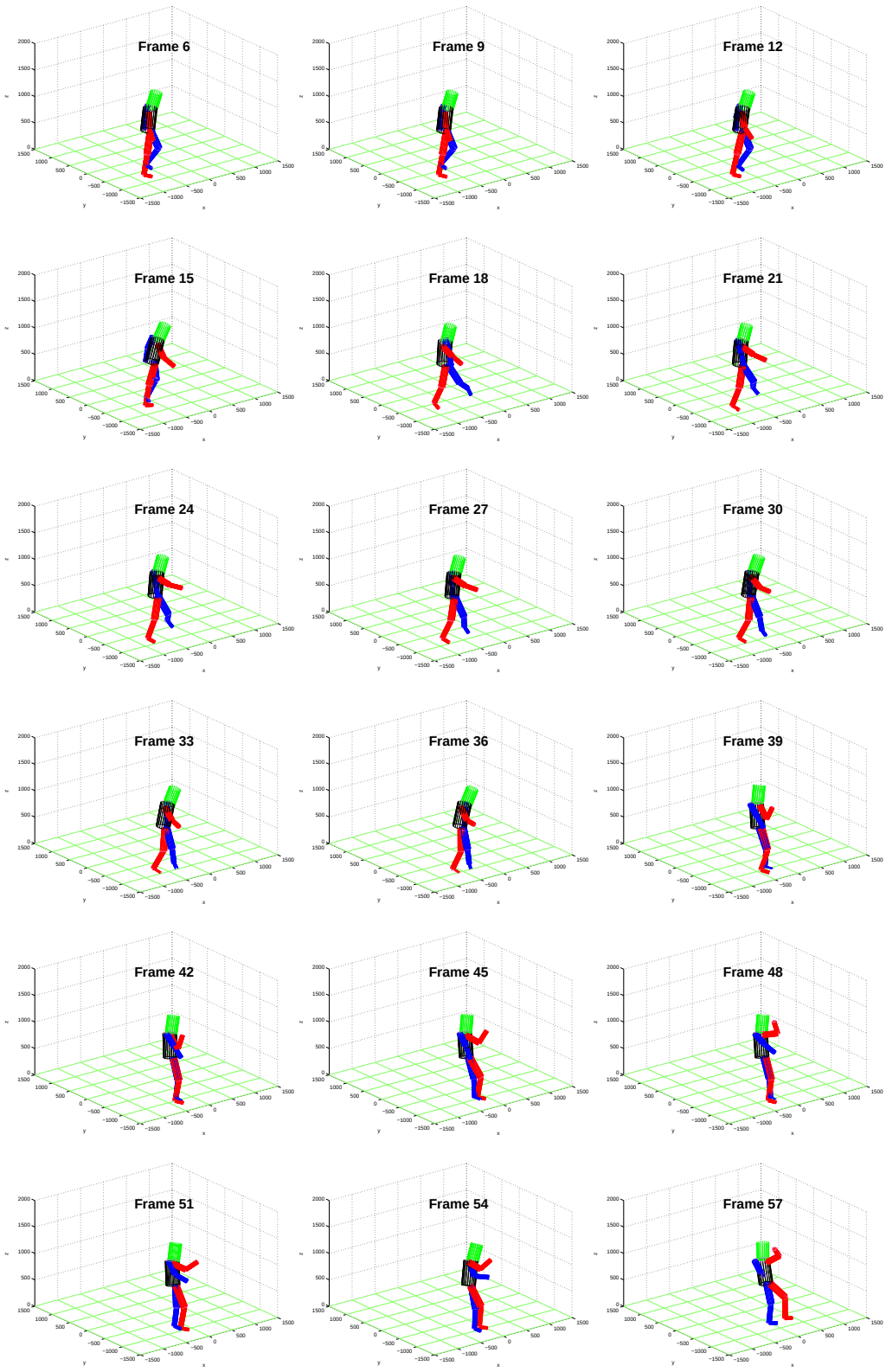



Figure 5.6: APF tracking. The 3D reconstruction of the *S1 Walking 1* sequence

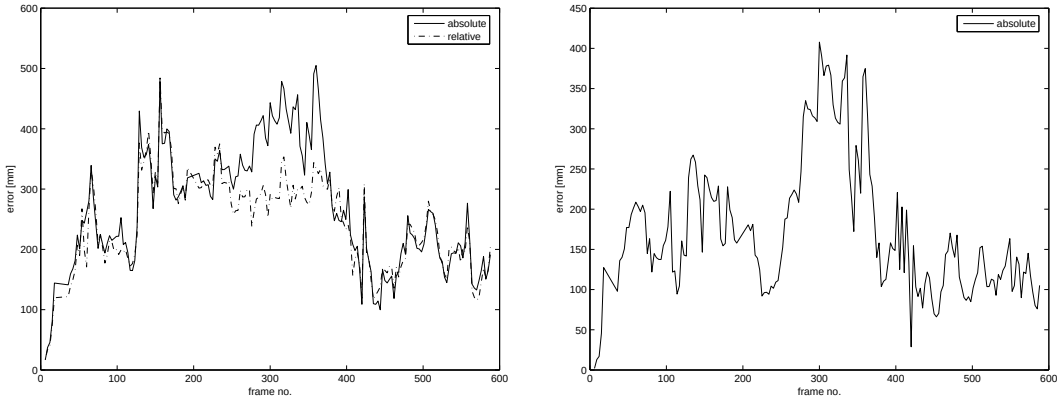



Figure 5.7: APF tracking: *S1 Walking 1* 3D error. The absolute body centre and the general position and pose errors (absolute and relative) are very high over the whole sequence, although they recover (to the level of the PF-SIR) at the end of the sequence.

The tracking results are inferior compared to the PF-SIR filter. Pose error steadily increases to values over 400mm, with high position errors (figure 5.7). These results are unexpected, since APF is claimed to perform better than the standard PF. However, figure 5.8 suggests that the variance of parameters $p^1...p^5$ increases through the multiple iterations of the APF (a similar pattern was found for $p^6...p^{24}$, not shown). Decreases are provoked by the initialisation on the first level, but additional levels result in particle spreading rather than concentration. Possible reasons behind this are the noisy likelihood compared to Deutscher's (who used high contrast background images) and the unstructured particle update that cannot converge with the limited particles and iterations.

Summarising, the APF fails to enhance tracking because the hierarchical structure of the parameters is ignored and they are not adjusted in the order of their hierarchical dependency. The physically related parameters (*e.g.* of a limb) are uncorrelated; the motion model is a general 0GM motion model, and therefore the generated poses are physically impossible or unlikely.

## 5.3 Hierarchical tracking

The hierarchical organisation of processes in the visual cortex (section 2.1.3) prompts a similar, multi-level, refinement based technique for an effective processing of the video input. The Hierarchical Partitioned Particle Filter (HPPF) has these properties, and recovers first the global and then the contingent parameters (*i.e.* a coarse to fine approach), while also modelling explicitly the physiological independence of some parameters.

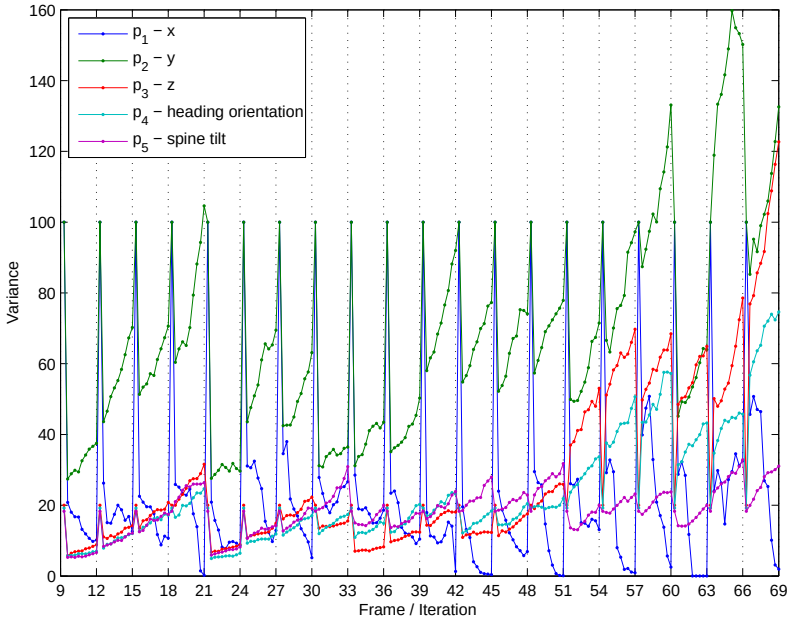



Figure 5.8: APF variance evolution. Scaled variance of the first five parameters (3D position, heading orientation, spine angle) for frames 9 to 66, each separated by the vertical grid, and for their 10 annealing levels, between the grid.

The difficulty of the human tracking arises from the complex structure of the human body, which implies a high dimensional space, and from the limited data available. To mitigate the slow convergence in this space, HPPF directs it by a prior know inter-dependence and independence of some parameters. After the presentation of the architecture and the generic algorithm, the HPPF is customised for AHHM tracking. However, it is generic and with appropriate partition and level definition is applicable to other structures.

### 5.3.1 Architecture

HPPF can be viewed as a mixture of the APF and the PPF, though there are important differences. The earlier work takes no account of the hierarchical dependency fixed by the structure of the tracked object. Further, for faster convergence and to avoid local minima, HPPF allows specialised priors and likelihoods for both partitions and levels.

#### The hierarchy

When the parameters are inter-dependent, it is natural to adapt the PF to a hierarchy in which a particle goes through levels of filtering for each new observation; similar to APF, each adjusts a sets of inter-dependent parameters, while other parameters are only slightly

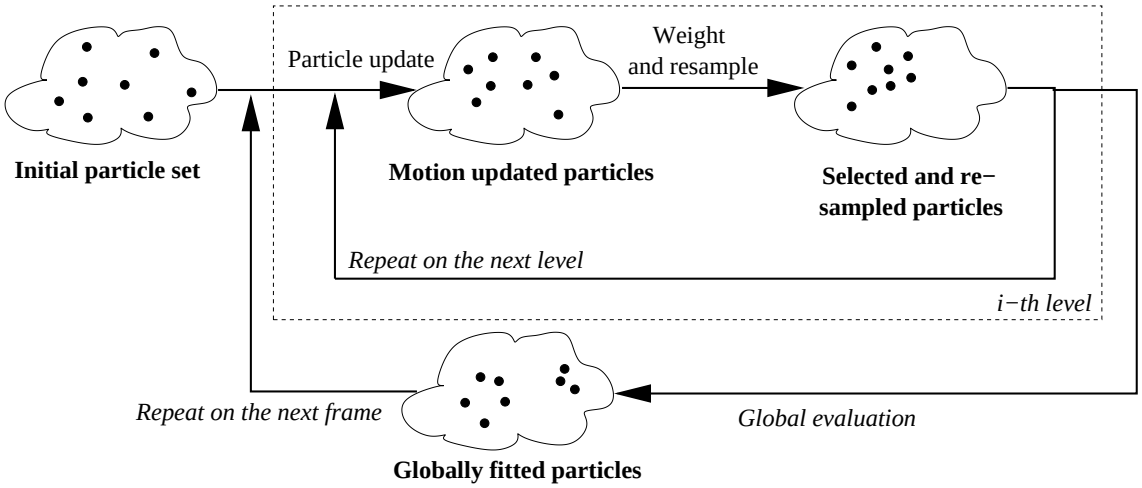



Figure 5.9: HPPF particle set on multiple levels, followed by global evaluation and resampling. On each level particles are updated by a probabilistic prior motion model, evaluated by the likelihood function and resampled.

affected if at all.

Over each level, the set of operations is identical, therefore levels are processed iteratively. With the evolution of the particle set, figure 5.9 suggests this structure. Each level consists of the basic PF phases: propagation, evaluation and resampling.

The final *global evaluation* and re-sampling keeps the best global solutions generated from the mixture of good local solutions and computes the estimate. Since it needs an evaluation and resampling, it is considered an additional level, but without a motion model that alters previous level particles.

### The partitions

Sets of parameters on a level might be independent similar to the PPF. Therefore these parameter partitions are evaluated and propagated separately. As for the AHHM, the parameters on level two and three (*i.e.* of the four limbs) are physically unconnected. However, their independence is arguable, since for some activities (*e.g.* walk, jog) the limbs are highly correlated, but not for an unrestricted activity. The MCM in section 5.3.6 will account for this. Partitioning exists also both between levels, adjusting different groups of parameters; and on each level, adjusting independent parameters in parallel.

**Algorithm 10**: HPPF Algorithm

**Input**: $\Psi_{t-1} = \{\mathrm{p}_{t-1}(i)\}_{i=1}^{n_p}$ – previous particle set and
$\mathrm{O}_t$ – current observation
**Output**: $\Psi_t$ – current particle set
$\overline{\mathrm{p}_t}$ – state estimate

1 ${}^0\Psi_t = \Psi_{t-1}$
2 **for** $i = 1 : n_p$ **do**
3 ${}^0\mathrm{p}_t(i)$ =`UpdateHistory`$({}^0\mathrm{p}_t(i))$ `// updates particle history, optional`
4 **end**
5 **for** $l = 1 : n_L$ **do** `// for each of the` $n_L$ `levels`
6 **for** $i = 1 : n_p$ **do** `// all particle propagate`
7 ${}^l\mathrm{p}_t(i) \sim q_l({}^l\mathrm{p}_t(i)|{}^{l-1}\mathrm{p}_t(i), l)$ `// details in section 5.3.6`
8 **end**
9 **for** $\Phi \in Partitions_l$ **do** `// for each partition on level` $l$
10 **for** $i = 1 : n_p$ **do**
11 $\tilde{w}_t^\Phi(i) = \pi_l^\Phi(\mathrm{p}_t(i)) \cdot \lambda_l^\Phi(\mathrm{O}_t|\mathrm{p}_t(i))$ `// importance weights from prior`
`// probability and likelihood (details in section 5.4.4)`
12 **end**
13 **end**
14 $\breve{w}_t^\Phi(i)$=`EnhanceCloud`$(\tilde{w}_t^\Phi(i))$ `// details in section 5.4.6`
15 **for** $\Phi \in Partitions_l$ **do** `// for each partition on level` $l$
16 $t = \sum_{i=1}^{n_p} \breve{w}_t^a(i)$ `// normalise total weights`
17 **for** $i = 1 : n_p$ **do**
18 $w_t^\Phi(i) = \frac{\breve{w}_t^\Phi(i)}{t}$
19 **end**
20 **end**
21 **if** $l = n_L$ **then** `// if global level then compute estimate`
22 $\overline{\mathrm{p}}_t$ = `ComputeEstimate`$(\{{}^l\mathrm{p}_t(i), w_t^\Phi(i)\}_{i=1}^{n_p})$ `// details in section 5.4.3`
23 **end**
24 $\{{}^l\mathrm{p}_t(i), -\}_{i=1}^{n_p}$ = `Resample`$(\{{}^l\mathrm{p}_t(i), w_t^\Phi(i)\}_{i=1}^{n_p}, l)$ `// generalised resampling`
25 **end**
26 $\Phi_t = \{{}^{n_L}\mathrm{p}_t(i)\}_{i=1}^{n_p}$ `// new particle set uses the last level particles`

### 5.3.2 Algorithmic description

The HPPF, algorithm 10, resembles the basic PF from section 5.2.3, however the multiple levels and partitions require modification. First, the particle evolution (in lines 5–25) is multiplied on $n_L$ levels. Since the the local and global levels are similar, the global level is considered as an additional level, the $n_L$-th, of the HPPF, with the procedure in lines 21–23 computing the PF estimate from the final particle set. Note that this precedes the final resampling (line 24), for the reason that with a limited number of particles the resampling might alter the actual particle distribution.

The operations from the loop of figure 5.9 are standard for a PF. The *particle update*,

lines 6–8, applies the level-dependent prior distribution $q_l$, given the motion model. For *weight* and *global evaluation*, lines 9–13 assess each particle partition independently with the level and partition dependent likelihood $\lambda_l^\Phi$ and the weighting prior probability $\pi_l^\Phi$.

The HPPF mechanism is conditioned mainly by the propagation prior (in line 7) and the weighting prior and likelihood (in line 11). Further, the `EnhanceCloud` function can alter the particle cloud in order to enhance the distribution, similar to the elimination applied for the PF-SIR. Possible mechanisms are discussed in section 5.4.6. The `ComputeEstimate` function estimates the state for time $t$. This is generally the expectation of all particles

$$\overline{\mathrm{p}}_t = E < \mathrm{p}_t(i) >, \tag{5.29}$$

while further alternatives are explored in section 5.4.3. Finally, the `UpdateHistory` (line 3) maintains the particle history for the movement modelling particle from section 5.3.6.

**Algorithm 11**: Generalised resampling for HPPF

**Input**: $\{\mathrm{p}_t(i), w_t^\Phi(i)\}_{i=1}^{n_p}$ – particle set
$l$ – current level
**Output**: $\{\hat{\mathrm{p}}_t\}_{j=1}^{n_p}$

1 $\overline{\mathrm{p}}_t = \mathrm{p}_t$ `// initialise. not affected parameters stay unchanged`
2 **for** $\Phi \in Partitions_l$ **do** `// for each partition on level` $l$
3 $c_1 = w_t^\Phi(1)$ `// initialise cumulative sum of weights (CSW)`
4 **for** $i = 2 : n_p$ **do**
5 $c_i = c_{i-1} + w_t^\Phi(i)$ `// construct CSW`
6 **end**
7 $i = 1$ `// start from sampling`
8 $u_1 \sim \mathcal{U}[0, {n_p}^{-1}]$ `// draw starting point`
9 **for** $i = 1 : n_p$ **do**
10 $u_j = u_1 + {n_p}^{-1}(j-1)$ `// next sample`
11 **while** $u_j > c_i$ **do**
12 $i = i + 1$
13 **end**
14 $\hat{\mathrm{p}}_t^\Phi(j) = \mathrm{p}_t^\Phi(i)$ `// re-sample`
15 **end**
16 **end**

The recombination of partitions is part of the particle re-sampling. The HPPF repeats the generalised resampling for every partition, algorithm 11 of the SIR (section 5.2.3).

Particles are re-sampled on a per partition basis, in parameter sets, weighted by the likelihood of their partition. Lower and higher level parameters inherently can mix and

result in a global configuration with low likelihood. However, this is partly balanced by slight adjustments of the higher level parameters at a lower level, assuring their stability and avoiding degradation through mixing. This combination is essentially the same as the crossover operation between particles that is considered favourable for the APF [117].

The special cases of the HPPF are:

- one level, $n_L = 1$, and one partition, $|Partitions_1| = 1$ is equivalent with the SIR filter;
- multiple levels, $n_L = 1$, and one partition, $|Partitions_l| = n$ is equivalent with the partitioned particle filter;
- multiple levels, $n_L = m$, and one partition, $|Partitions_l| = 1$, $i = 1..m$ is equivalent with the annealed particle filter

### 5.3.3 HPPF for AHHM

The four levels ($n_L = 4$) of the HPPF applied to the human body model are shown in figure 5.10, and stated explicitly in table 5.2. On the first level the most independent parameters (*i.e.* torso global position) are adjusted, followed by the hierarchically inferior parameters (*i.e.* the upper then the lower limbs). The resulting particles from level three are re-evaluated and the best global fits provide the final particle distribution. Partitions are independent sets of dependent limb parameters.

Particularly for AHHM, HPPF updates each partition with a 0GM. Global, edge, silhouette and colour likelihoods from section 3.3 are used along with the range (section 3.2.2) and the maximum visibility priors (section 3.2.6) as importance weights. `EnhanceCloud` performs the weight enhancement briefly introduced for PF-SIR.

### 5.3.4 Quantitative evaluation of PFs for AHHM

The above HPPF is evaluated quantitatively against the PF-SIR and APF, previously presented. For comparability, $n_p = 150$ particles are used, which for three local and one global level results in an equivalent number of particles (*i.e.* likelihood evaluations) to the SIR-PF.

Table 5.3 compares the 3D and 2D relative and absolute errors for the seven test sequences (table 5.1). The results show that the APF performs more poorly than the

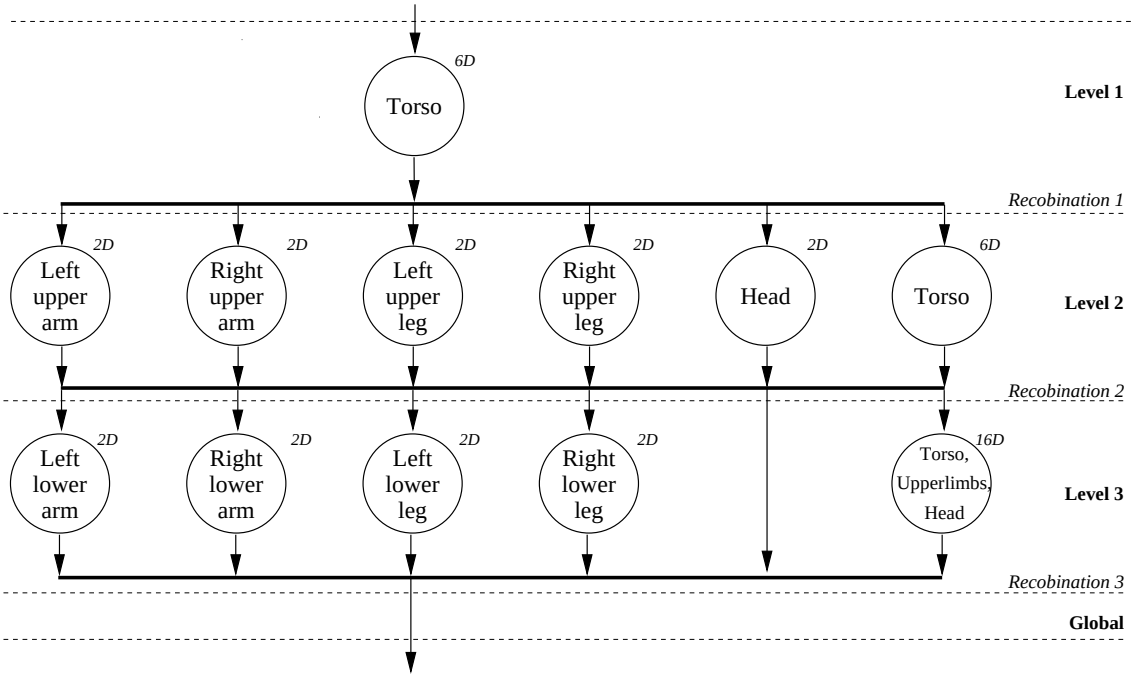



Figure 5.10: HPPF for human tracking. The first level modifies the global, torso parameters; followed by the upper limb and head, and finally by the lower limb parameters. Each body part forms a partition, while the last partition on levels two and three contains higher level parameters accommodated only slightly to fit the lower level body parts. The *global* level evaluates the whole particle and provides the next generation of particles.

| **Level** | **Description** | **Number of partitions** | **Name of partition** | **PV partition** |
|---|---|---|---|---|
| $l$ | | $\lvert Partitions_l \rvert$ | $Partitions_l$ | $\Phi \in Partitions_l$ |
| 1 | position | 1 | torso | $p^1, \ldots, p^6$ |
| 2 | upper limbs | 6 | left upper arm | $p^9, p^{10}$ |
| | | | right upper arm | $p^{13}, p^{14}$ |
| | | | left upper leg | $p^{17}, p^{18}$ |
| | | | right upper leg | $p^{21}, p^{22}$ |
| | | | head | $p^7, p^8$ |
| | | | torso | $p^1, \ldots, p^6$ |
| 3 | lower limbs | 5 | left lower arm | $p^{11}, p^{12}$ |
| | | | right lower arm | $p^{15}, p^{16}$ |
| | | | left lower leg | $p^{19}, p^{20}$ |
| | | | right lower leg | $p^{23}, p^{24}$ |
| | | | torso, upper limbs, head | $p^7, p^8$ |
| 4 | global | 1 | all | $p^1, \ldots, p^{23}$ |

Table 5.2: Partition definition per level, corresponding to figure 5.10, with each partition's PV parameters are listed (table 3.9)

PF-SIR, confirming our previous visual observation. For the S3 *Gesture* 1 sequences, the recovered model does not project on to the visible image, therefore its 2D error is huge, explaining the high camera C2 error. The elimination with `EnhanceCloud` reduces the SIR error, while HPPF further mitigates it. This validates the effectiveness of the hierarchical-partitioned approach.

| Method | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|
| | rel | abs | rel | abs | rel | abs |
| PF-SIR (no elimination) | 146.0 | 152.5 | 25.2 | 27.2 | 23.6 | 24.1 |
| PF-SIR (with elimination) | 137.6 | 141.7 | 23.9 | 25.3 | 23.1 | 23.7 |
| APF | 317.6 | 434.5 | 53.6 | 72.5 | $\infty$ | $\infty$ |
| HPPF | 92.0 | 97.6 | 17.6 | 19.8 | 18.2 | 18.7 |

Table 5.3: Particle filter variant accuracy. 3D and 2D (camera C1 and C2) relative and absolute errors for PF with 0GM, APF and HPPF with 0GM

| Method | Particles $n_p$ | Levels $n_L$ | Time / frame [s] | Time / particle [ms] |
|---|---|---|---|---|
| PF-SIR (no elimination) | 600 | 1 | 10.6 | 17.7 |
| PF-SIR (with elimination) | 600 | 1 | 10.3 | 17.2 |
| APF | 60 | 10 | 10.8 | 17.9 |
| HPPF | 150 | 3 | 10.5 | 17.4 |

Table 5.4: Particle filter variant speed. For each test, the likelihood evaluation was kept constant at 600 evaluations/frame, given by $n_p \times n_L$.

The execution times are shown in table 5.4. For all algorithms, the total number of likelihood evaluations are identical, equivalent for the case of $n_p = 600$ particles used in the PF-SIR filter. The execution time does not include the input image pre-processing and the visualisation. The evaluation was performed on a workstation with a 3.2GHz Pentium 4 CPU, 2.5GB of RAM, using only a single core.

The execution times are quasi-equal. This shows that level based processing does not have an overhead and the time complexity of the above PF algorithms depends only on the total number of likelihood evaluations and indirectly on the number of particles.

In the next section, the MCM will further enhance these tracking results by adding the prior dynamics model learnt in chapter 4, while further alternatives and comparison of these and their parameters are also given.

### 5.3.5 HPPF and other PFs

The similarity of HPPF to APF and PPF was shown in section 5.3.2. HPPF can be seen as a generalisation of the two. NBP [135] and PAMPAS [138] (see section 2.2.2) also apply a partitioned tracking, however in these body parts are not hierarchically dependent. Therefore, computational effort might be wasted in recovering independent parts that do not assemble into a body. HPPF, like visual processing, also has backward propagation of the inferred knowledge, by tuning higher level parameters on a lower level.

The Subspace Hierarchical Particle Filter of Brandão *et al.* [224] is very similar to HPPF, however the fixed structure of HPPF incorporates explicitly the anatomical prior knowledge, with the price of lost generality after object structure is chosen. In contrast, the advantages of HPPF tracking are the partition specific motion priors and likelihoods, and the more complex object that can be considered (*i.e.* human compared to hand).

### 5.3.6 HPPF with MCM

The Movement Cluster Model (MCM) from chapter 4 offers a detailed dynamic model for articulated human motion: it allows continuous pose parameters, a motion history of $l_m > 1$, and, unlike CTPTM, fits instantaneous transition definitions required by the PF.

Although CTPTM has potential for synthetic motion generation, it is incompatible with tracking by HPPF. If two partitions partially overlap and the transitions have different timing parameters, their combination, the resampling, into a single pose is impossible.

However, MCM requires the history of movements. Therefore the particle $\mathrm{p}_t$ will model a movement and not a pose. The particle $\mathrm{p}_t$ consists of a sequence, starting from the current pose at time $t$, pose ${}_0\mathrm{p}_t$ and its $l_m$ long history ${}_1\mathrm{p}_t, {}_2\mathrm{p}_t, \ldots, {}_{l_m-1}\mathrm{p}_t$ at $t-1, \ldots, t-l_m$. Further, the previous movement with BFV of the partition $\phi$, is $\mathrm{m} = [{}_{l_m}\mathrm{p}_t^{\phi}, \ldots, {}_1\mathrm{p}_t^{\phi}]$.

Then, the `UpdateHistory` function from line 3 of the HPPF algorithm performs the time update by shifting the BFVs in the particle $\mathrm{p}_t(i)$:

$$\texttt{UpdateHistory}(\mathrm{p}_t(i)) = [{}_{-l_m+1}\mathrm{p}_{t-1}(i), \ldots, {}_0\mathrm{p}_{t-1}(i), {}_0\mathrm{p}_{t-1}(i)], \tag{5.30}$$

and prepares to generate the new, propagated pose ${}_0\mathrm{p}_t(i)$.

The MCMs provide multiple possibilities for particle propagation. First, by their multiple detail levels, five propagation types are defined in table 5.5 using different MCMs: *none*, with no updated pose parameter; *pose*, with all model parameters simultaneously

updated; *limb*-based, with individual limb parameters changed simultaneously, but different limbs in parallel; *independent*, limb update, with independent lower and upper limbs; and *lower limb*, with only inferior limbs updated. For each propagation type, the sets of MCMs involved, together with their propagated parameter partitions, are shown in table 5.5.

| **Type** | **Description** | **Number of MCMs** | **MCMs** | **Parameter partition ($\phi$)** |
|---|---|---|---|---|
| none | no pose | 0 | | |
| pose | whole pose | 1 | $\mathcal{M}_1$ | $p^5, p^6, p^9, \ldots, p^{24}$ |
| limb | head and 4 whole limbs | 5 | $\mathcal{M}_2$ | $p^7, p^8$ |
| | | | $\mathcal{M}_3$ | $p^9, \ldots, p^{12}$ |
| | | | $\mathcal{M}_4$ | $p^{13}, \ldots, p^{16}$ |
| | | | $\mathcal{M}_5$ | $p^{17}, \ldots, p^{20}$ |
| | | | $\mathcal{M}_6$ | $p^{21}, \ldots, p^{24}$ |
| independent | head, 4 upper and 4 lower limbs | 9 | $\mathcal{M}_2$ | $p^7, p^8$ |
| | | | $\mathcal{M}_7$ | $p^9, p^{10}$ |
| | | | $\mathcal{M}_8$ | $p^{13}, p^{14}$ |
| | | | $\mathcal{M}_9$ | $p^{17}, p^{18}$ |
| | | | $\mathcal{M}_{10}$ | $p^{21}, p^{22}$ |
| | | | $\mathcal{M}_{11}$ | $p^{11}, p^{12}$ |
| | | | $\mathcal{M}_{12}$ | $p^{15}, p^{16}$ |
| | | | $\mathcal{M}_{13}$ | $p^{19}, p^{20}$ |
| | | | $\mathcal{M}_{14}$ | $p^{23}, p^{24}$ |
| lowerlimb | 4 lower limbs | 4 | $\mathcal{M}_{11}$ | $p^{11}, p^{12}$ |
| | | | $\mathcal{M}_{12}$ | $p^{15}, p^{16}$ |
| | | | $\mathcal{M}_{13}$ | $p^{19}, p^{20}$ |
| | | | $\mathcal{M}_{14}$ | $p^{23}, p^{24}$ |

Table 5.5: Propagation type definition. For each *type*, the number and the list of MCMs defining the update is specified, together with the PV partition of the MCM (based on table 4.3)

For hierarchical coarse to fine processing, the propagation type is HPPF level depen-

dent, and it was chosen a priori with the probabilities:

$$\mathcal{P}(type = none|l = 1) = 1, \tag{5.31}$$

$$\mathcal{P}(type = limb|l = 2) = 0.34, \tag{5.32}$$

$$\mathcal{P}(type = pose|l = 2) = 0.33, \tag{5.33}$$

$$\mathcal{P}(type = independent|l = 2) = 0.33, \tag{5.34}$$

$$\mathcal{P}(type = lowerlimb|l = 3) = 1. \tag{5.35}$$

Equations 5.32–5.35 fix *type* = *none* on the first and *type* = *lowerlimb* the third levels, while on the second level assumes equal probabilities for *type* = *limb*, *pose* and *independent* propagation types.

Therefore, on level one no pose parameters are updated, but only the 3D position and global orientation. On level two either the pose, the limb or the independent model is used, with similar probabilities. On the third level, all parameters are fixed, except the lower limb parameters, most dependent on the other body parts. This propagation strategy fits the hierarchical and partitioned design of the HPPF, since at the highest level, the most independent parameters are changed, then they are fixed, while the lower level parameters are adjusted; also, parameters are updated with the independent partitions of the MCM.

The propagation, algorithm 12, on all levels, except the last, global one, updates the previous particle set, processing particles individually. The update depends on the current tracking level, the propagation *type* and on the random motion *mode*. The propagation convergence is not theoretically proved, however experiments suggests that with this propagation the HPPF tracks the target over the whole test sequences. Also, it can be argued that this propagation is a systematic cross-over operation successfully applied with the APF by Deutcher and Reid [117].

The global parameters (position and orientation), are updated according to a 0GM in line 6, with noise variance $\mathbf{P}_G$ high on $l = 1$ and low on the other levels.

Then, for the selected propagation, each involved MCM updates distinct particle partitions $\phi$. Algorithm 5 from chapter 4 performs this, with one of the four modes (**pose**, **randompose**, **speed**, **normal**), defined in section 4.5.2.

**Algorithm 12**: Propagation with MCM

**Input**: p – current particle
$l$ – current level
**Output**: p̌ – next particle

p̌ = p
**if** $l = n_L$ **then** `// if global level`
  `exit` `// no update is needed`
**end**
$\mathrm{w} \sim \mathcal{N}(\mathrm{w}; 0, \mathbf{P}_G)$
$\check{\mathrm{p}}^{global} = \mathrm{p}^{global} + \mathrm{w}$ `// global parameter update`
$type \sim \mathcal{U}\{type; (\mathcal{P}(type|l), type)\}$ `// sample with equations (5.32)-(5.35)`
**for** for each MCM $\mathcal{M}_i$ for propagation $type$ from table 5.5 **do**
  $\phi = \mathcal{M}_i.\phi$ `// partition of` $\mathcal{M}_i$ `(table 5.5)`
  $mode \sim \mathcal{U}\{motion; (p_{\mathbf{p}}, \mathbf{pose}), (p_{\mathbf{r}}, \mathbf{randompose}), (p_{\mathbf{s}}, \mathbf{speed}), (p_{\mathbf{n}}, \mathbf{normal})\}$
  $\check{\mathrm{p}}^{\phi}$= `GetNextBFV(`$\mathcal{M}_i$, *mode*, $\mathrm{p}^{\phi}$`))` `// generate new BFV for the particle`
  `// with algorithm 5 from chapter 4`
**end**

For optimum propagation, the effect on tracking of these modes is analysed in five test cases from table 5.6. The 0GM ($mode$ = **normal**), similar to section 5.3.3, ignores any learnt motion structure, and assumes the next pose within the learnt Gaussian variance $Model.BFV.\mathbf{P}$ (section 4.5.3) close to previous pose.

| Method | Mode selection probabilities | | | | 3D error [mm] | |
|---|---|---|---|---|---|---|
| | $p_{\mathbf{p}}$ | $p_{\mathbf{r}}$ | $p_{\mathbf{s}}$ | $p_{\mathbf{n}}$ | rel | abs |
| 0GM | 0.0 | 0.0 | 0.0 | 1.0 | 102.7 | 103.6 |
| MCM pose only | 1.0 | 0.0 | 0.0 | 0.0 | 120.1 | 119.9 |
| MCM pose and speed | 0.5 | 0.0 | 0.5 | 0.0 | 102.3 | 102.5 |
| MCM pose and 0GM | 0.5 | 0.0 | 0.0 | 0.5 | 111.6 | 113.3 |
| MCM pose, random pose, speed and 0GM | 0.3 | 0.1 | 0.3 | 0.3 | 89.1 | 91.6 |

Table 5.6: Errors of HPPF for MCM propagation modes: 3D relative and absolute errors for NBFV generation

The MCM pose only ($mode$ = **pose**) is the poorest, since recovery from errors is hard with the strong motion priors. Further, the initialisation, which assumes poses before the first frame identical to the first, also provokes initial errors. MCM pose and speed provides similar results to 0GM, provided by the higher chance of recovery from initial and intermediate failures.

The tracking error is the lowest if these three propagation modes are combined, and also with the additional, low probability, $mode$ = **randompose** to facilitate quick, unexpected

pose transition or recovery from failure.

The propagation also depends on the heating parameters, considered fixed in the above tests with $\sigma_P = \sigma_L = \sigma_S = 1$ and $\sigma_N = 0.5$. Later, the effect of these parameters is tested in section 5.4.5.

The figures suggest that tracking is more accurate when a mixture of update modes is used.

### 5.3.7 Model complexity

The articulated human model from chapter 3 adds large complexity to a blob model. Thanks to the structured design and coding of the HPPF, the adaptation of the tracker for other models is straightforward. To reduce the complexity, for example to remove the arms, only the partition definition from figure 5.10 and table 5.2 has to be changed, by removing partitions corresponding to the arms. This implicitly reduces the particle dimensionality and the complexity of update and evaluation in the HPPF.

Several frames of the simplified tracker's result are shown in figure 5.11. Table 5.7 suggests about 32% relative error decrease. This is partly due to the removed largeerror components of the arm, and partly due to the imporved particle number per particle dimension ratio.

| Tracked | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|
| | rel | abs | rel | abs | rel | abs |
| Trunk, legs and head tracking | 64.4 | 72.4 | 18.1 | 19.3 | 17.7 | 18.2 |
| Full body tracking, error without arms | 65.2 | 73.4 | 17.8 | 19.0 | 17.9 | 18.3 |
| Full body tracking, full error | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |

Table 5.7: Errors for simplified model human model. Compared to the full articulated human model, tracking only lower limbs, trunk and head reduces the 3D and 2D, relative and absolute errors. Lines two and three show that the the error reduction results mainly from the ignored limb errors, however the reduced complexity tracker further improves this error

## 5.4 Parameter adjustment for HPPF with MCM

Both the hierarchical filter and the MCM depend on a set of parameters. As is usual, tracking accuracy depends on whether these parameters are optimised. Therefore, to find their optimal value, this section systematically analyses them. Parallel full optimisation

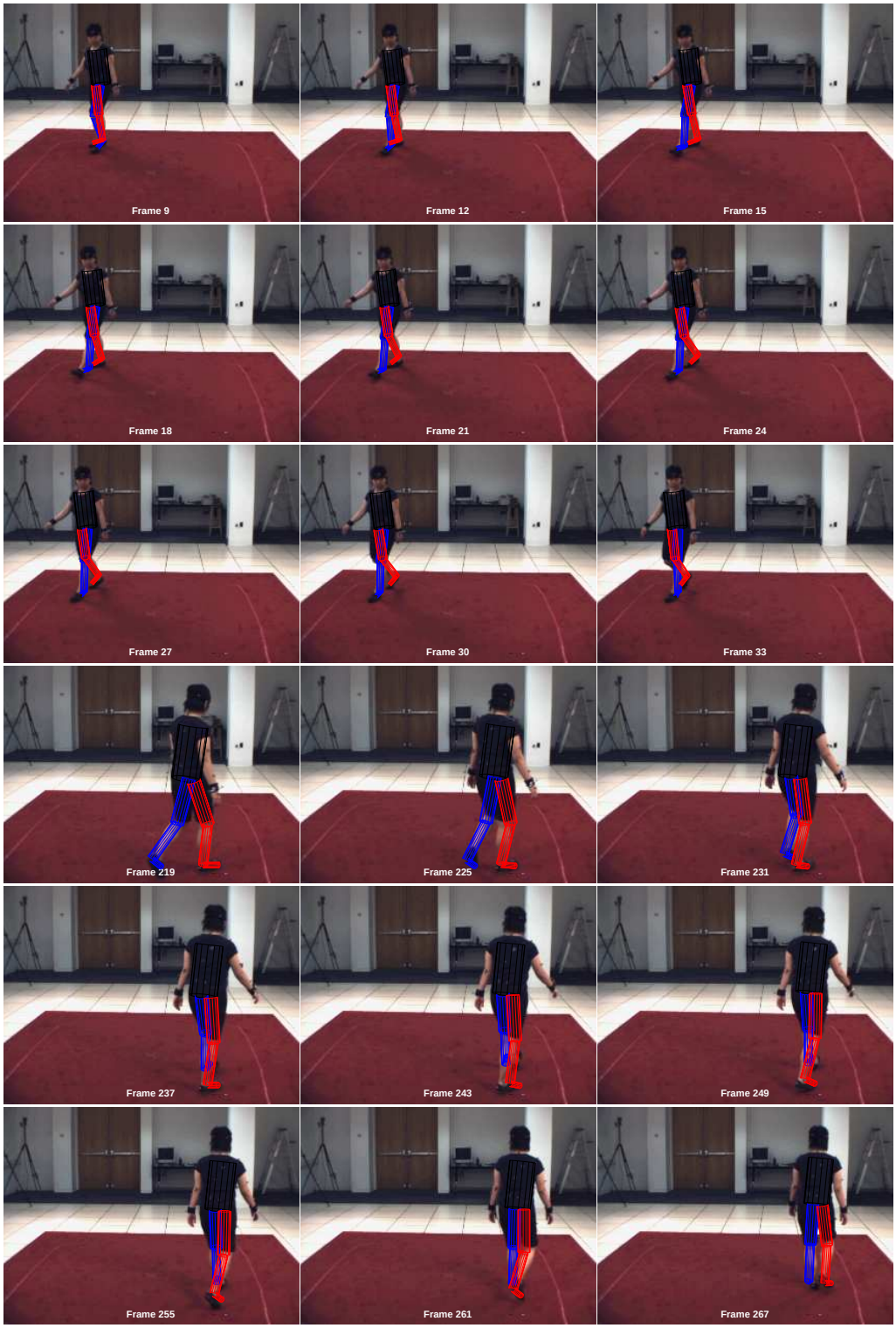



Figure 5.11: Reduced complexity human model tracked with the HPPF in the *S1 Walking 1* camera C1 sequence. Tracking results for frames 9–33 and 219–267 are superimposed with the input image [◊].

of all parameters is not possible, because of the combinatorial increase of the possible test cases. Thus, parameters are altered empirically, either individually or in related sets, to evaluate and maximise their tracking performance. The optimal values are summarised at the end of the section (table 5.15) and, where it is possible, these values are used to test the effect of other parameters. Hence, experiments have been repeated iteratively, optimising different parameters. However, here only the final parametrisation with optimal or suboptimal fixed parameters is presented for each test case.

### 5.4.1 Motion model parameters

A low MCM movement length, $l_m$, results in short motion memory, while high $l_m$ means long motion memory. A low number of MC, $n_{\mathcal{C}}$, restricts the number of movements and therefore the estimation of the current movement is less accurate and the prediction of the next pose is inferior. Table 5.8 shows the effects of both $n_{\mathcal{C}}$ and $l_m$. These are not straightforward, but it can be seen in the table that for $n_{\mathcal{C}} = \{20, 40, 60, 80, 100\}$, the lengths of movement with minimum errors are for $l_m = \{35, 15, 25, 5, 1\}$. Similarly, for $l_m = \{1, 3, 5, 15, 25, 35\}$ the best numbers of clusters are $n_{\mathcal{C}} = \{100, 40, 80, 40, 60, 20\}$. Though $n_{\mathcal{C}} = 40$ is irregular, the other sequences show the opposing effects of the two parameters, matching the conclusions of chapter 4.

Therefore, the global minimum is expected at the middle of the range of both parameters, and from table 5.8, the minimum error is obtained for $n_{\mathcal{C}} = 80$ and $l_m = 5$.

### 5.4.2 Number of particles

The distribution of pose space is represented by the set of particles. The number of particles grows exponentially with the dimensionality of the parameter vector [117, 128]. Therefore PFs perform better with increased $n_p$, however this increases the processing time.

Table 5.9 confirms the increase of performance, although above $n_p = 150-200$ particles, no significant performance enhancement can be observed, and the processing time overhead is not motivated. Setting $n_p = 150-200$ particles, on the global and the three local levels, results in a total of $600 - 800$ likelihood evaluations, lower than the 1000 evaluations suggested for good performance of APF [117], and it could be further reduced to $4 \times 85 = 340$ evaluations ($n_p = 85$), if 11% higher absolute 3D error is tolerated.

| $n_{\mathcal{C}}$ | $l_m$ | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|---|
| | | rel | abs | rel | abs | rel | abs |
| 20 | 1 | 108.5 | 110.3 | 19.8 | 21.2 | 19.5 | 20.0 |
| | 3 | 98.1 | 99.6 | 18.3 | 19.8 | 18.9 | 19.1 |
| | 5 | 94.0 | 94.4 | 17.8 | 19.2 | 18.0 | 18.3 |
| | 15 | 99.4 | 101.7 | 18.3 | 19.9 | 18.9 | 19.4 |
| | 25 | 99.4 | 101.7 | 18.3 | 19.9 | 18.9 | 19.4 |
| | 35 | 91.5 | 92.9 | 17.4 | 18.7 | 17.8 | 18.2 |
| 40 | 1 | 95.5 | 97.5 | 18.4 | 19.7 | 18.3 | 18.8 |
| | 3 | 93.7 | 95.2 | 18.3 | 19.8 | 17.8 | 18.1 |
| | 5 | 103.1 | 105.4 | 19.2 | 20.7 | 18.9 | 19.3 |
| | 15 | 89.6 | 91.1 | 17.4 | 18.9 | 17.5 | 17.7 |
| | 25 | 121.4 | 122.5 | 21.1 | 22.5 | 21.8 | 21.9 |
| | 35 | 115.5 | 115 | 20.4 | 21.7 | 20.6 | 20.6 |
| 60 | 1 | 102.3 | 102 | 19.0 | 20.2 | 19.2 | 19.2 |
| | 3 | 114.1 | 115 | 20.1 | 21.5 | 20.5 | 20.7 |
| | 5 | 96.0 | 99.2 | 18.0 | 19.6 | 18.7 | 19.0 |
| | 15 | 100.7 | 101.8 | 18.6 | 20.2 | 19.3 | 19.6 |
| | 25 | 89.2 | 90.2 | 17.6 | 18.9 | 17.1 | 17.4 |
| | 35 | 101.8 | 102.1 | 19.2 | 20.7 | 18.8 | 18.8 |
| 80 | 1 | 102.1 | 104.4 | 19.0 | 20.5 | 19.3 | 19.7 |
| | 3 | 94.6 | 97.0 | 18.4 | 20.0 | 17.7 | 18.2 |
| | 5 | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| | 15 | 111.2 | 111.3 | 20.3 | 21.5 | 19.6 | 19.9 |
| | 25 | 105 | 105.1 | 19.5 | 20.9 | 19.4 | 19.6 |
| | 35 | 90.9 | 94.3 | 18.0 | 19.8 | 17.8 | 18.2 |
| 100 | 1 | 92.2 | 95.5 | 17.8 | 19.7 | 18.2 | 18.8 |
| | 3 | 98.1 | 100.4 | 19.0 | 20.6 | 18.3 | 18.7 |
| | 5 | 100.5 | 101.6 | 18.6 | 19.9 | 19.2 | 19.5 |
| | 15 | 104.1 | 104.0 | 19.1 | 20.3 | 19.2 | 19.3 |
| | 25 | 98.0 | 100.2 | 18.8 | 20.3 | 18.6 | 19.0 |
| | 35 | 94.3 | 96.7 | 18.4 | 19.9 | 17.6 | 18.2 |

Table 5.8: Tracking error dependence on $n_{\mathcal{C}}$ and $l_m$. 3D and 2D (camera C1 and C2) relative and absolute errors.

The errors per test sequences, shown in figure 5.12, suggest that *S1 Gesture 1* has the lowest errors, being trained and mostly static. The particle number is not relevant, while for the other sequences, a high enhancement at the initial increase of $n_p$ is observed.

The methods from this chapter, were evaluated with tests on a single run of the HPPF. However, HPPF is stochastic algorithm and individual runs may result in variations of the tracking error. Therefore, mean, maximum and minimum errors over a large number of executions are better metrics to compare.

The tracking errors of *S1 Walking 1* sequence, tracked 100 times with the HPPF,

| $n_p$ | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|
| | rel | abs | rel | abs | rel | abs |
| 50 | 97.5 | 99.7 | 18.2 | 19.7 | 18.7 | 18.9 |
| 65 | 112.2 | 113.3 | 19.8 | 21.3 | 20.5 | 20.7 |
| 75 | 117.1 | 113.2 | 20.4 | 21.1 | 20.6 | 20.3 |
| 85 | 97.0 | 97.9 | 18.5 | 20.1 | 18.3 | 18.4 |
| 100 | 91.5 | 94.7 | 17.5 | 19.3 | 17.8 | 18.4 |
| 150 | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| 200 | 89.9 | 90.3 | 17.3 | 18.8 | 18.0 | 18.0 |
| 350 | 93.3 | 93.7 | 18.1 | 19.4 | 17.9 | 18.2 |
| 500 | 86.7 | 87.0 | 16.9 | 18.4 | 17.3 | 17.5 |
| 750 | 83.2 | 84.8 | 16.3 | 17.7 | 16.9 | 17.4 |
| 1000 | 83.9 | 85.5 | 16.5 | 17.9 | 17.2 | 17.6 |

Table 5.9: Tracking error dependence on the number of particles. 3D and 2D (camera C1 and C2) relative and absolute errors.

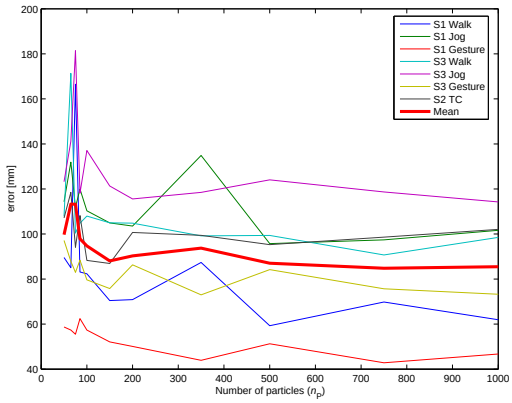
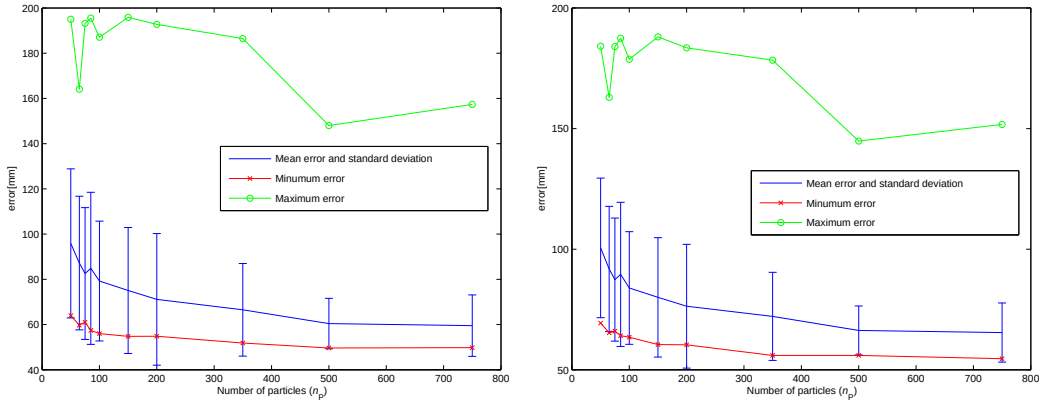



Figure 5.12: The effect of particle number on the 3D absolute error, for the seven individual HumanEva test sequences and for their mean.

from table 5.10 and figure 5.13, show expected tracking improvement with the increasing number of the particles. The previous paragraph argued for $n_p = 150$ particles. With more particles tracking error can be reduced further as table 5.10 shows. However, the current implementation due to the costly observation evaluation does not supports this increase. All errors, show decreasing tendency, except, since the minimum and maximum errors are defined only by a single run, their decline is not stable. The standard deviation of the mean error also decreases for up to $n_p = 500$ particles. The less than error 2mm increase for $n_p = 750$ can be motivated by the incompatibility of the tracked and the HumanEva models. Together with with high processing time, the tracking with $n_p = 750$ is not encouraged.

| $n_p$ | 3D relative error [mm] | | | | 3D absolute error [mm] | | | |
|---|---|---|---|---|---|---|---|---|
| | mean | std | max | min | mean | std | max | min |
| 50 | 95.89 | 32.99 | 195.04 | 63.95 | 100.54 | 28.94 | 184.12 | 69.37 |
| 65 | 87.19 | 29.54 | 164.09 | 59.71 | 91.78 | 25.99 | 162.96 | 65.32 |
| 75 | 82.56 | 29.14 | 193.22 | 61.02 | 87.40 | 25.51 | 183.99 | 66.02 |
| 85 | 84.87 | 33.63 | 195.50 | 57.44 | 89.56 | 29.88 | 187.47 | 64.15 |
| 100 | 79.23 | 26.48 | 187.06 | 56.01 | 83.92 | 23.35 | 178.72 | 63.51 |
| 150 | 75.08 | 27.85 | 195.84 | 54.77 | 80.03 | 24.75 | 188.03 | 60.44 |
| 200 | 71.14 | 29.11 | 192.76 | 54.81 | 76.34 | 25.65 | 183.49 | 60.36 |
| 350 | 66.51 | 20.49 | 186.48 | 51.85 | 72.14 | 18.27 | 178.37 | 55.96 |
| 500 | 60.43 | 11.15 | 148.03 | 49.65 | 66.32 | 10.14 | 144.83 | 55.93 |
| 750 | 59.52 | 13.59 | 157.37 | 49.77 | 65.43 | 12.27 | 151.74 | 54.60 |

Table 5.10: Errors statistics on the number of particles. The mean, standard deviation, minimum and maximum relative and absolute errors are shown for increasing number of particles.




(a) Relative error

(b) Absolute error

Figure 5.13: Particle number statistical analysis. For 100 repeated runs of the HPPF mean and standard deviation (in blue), minimum (in red) and maximum (in green) relative (a) and absolute (b) errors are shown for the *S1 Walking 1* sequence. With the increase of the number of particles the metrics improve.

A statistical evaluation as above is justified by evaluation robustness in all tests from this chapter. However, the running time of approximately 20 minutes per sequence, with 150 particles, on a 3.2GHz Pentium 4 CPU processor (single core is used only) is prohibitive. Therefore this test was performed only for the *S1 Walking 1* sequence only to evaluate the effect of particle number on the HPPF.

### 5.4.3 The tracking estimate

The tracker output is estimated in line 22 of the HPPF algorithm from the distribution described by the particle set. This estimate in the particle filter framework is generally the expectation [60, p.36] over the particle set, each particle weighted with its likelihood:

$$\overline{\mathrm{p}} = \sum_i w(i)\mathrm{p}(i). \tag{5.36}$$

For tracking and especially for human tracking, with a multi-modal particle distribution and with a limited number of particles compared to the dimensionality of the tracking space, the expected value of weighted particles drifts off towards a low probability configuration (figure 5.14a) therefore the estimate is distorted.

The global maximum a posteriori estimate (MAP), $\overline{\mathrm{p}} = \mathrm{p}_m$, where

$$m = \underset{i}{\operatorname{argmax}}\, w(i), \tag{5.37}$$

selects the best fitting particle and estimates the maximum mode of a particle set. If particles are dense, and therefore accurately characterise the distribution, then this is a good estimate. Because of its large dimensionality with sparse particles, the MAP estimate is frequently far from the distribution peak (figure 5.14b).

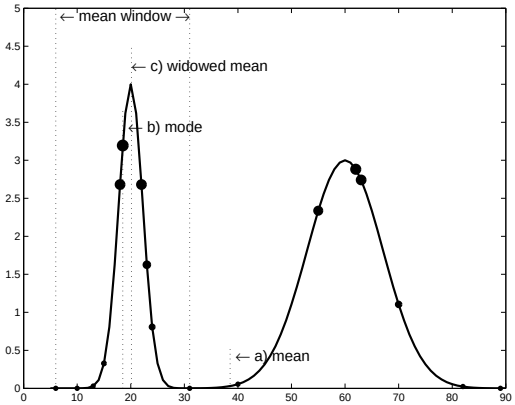



Figure 5.14: Tracking estimate: a) global mean, b) global MAP estimate, c) windowed-mean

The weighted mean of particles within a window around the MAP estimate, as also suggested by figure 5.14, is a more robust estimate since particles from other modes of the distribution are ignored.

For human tracking with the MCM, the window definition is given by the MC: a

particle is included in the estimate, if and only if it is in the same cluster as MAP particle. Therefore,

$$\overline{\mathrm{p}} = \frac{\sum_{\mathtt{GetMC}(\mathrm{p}_i)=\gamma} w(i)\mathrm{p}(i)}{\sum_{\mathtt{GetMC}(\mathrm{p}_i)=\gamma} w(i)}, \tag{5.38}$$

with the selected (equation (4.17)) MC:

$$\gamma = \mathtt{GetMC}(\mathrm{p}_m). \tag{5.39}$$

These three estimates are compared by their tracking error in table 5.11. The global mean has the highest error, while the combined Windowed-mean estimate is better then the MAP estimate. This is because the Windowed-mean estimate avoids misleading particles, and the larger number of particles gives a better estimate than the MAP estimate alone.

| Selection | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|
| | rel | abs | rel | abs | rel | abs |
| MAP estimate | 88.9 | 90.0 | 17.4 | 18.7 | 17.7 | 18.0 |
| Mean | 115.2 | 116.5 | 20.6 | 21.8 | 20.0 | 20.1 |
| Windowed-mean | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |

Table 5.11: Tracking error dependence on the estimation method. 3D and 2D (camera C1 and C2) relative and absolute errors.

### 5.4.4 Likelihoods and priors

Likelihoods are the only components of the tracker that provide the input from observations, from the video sequence. Therefore their careful design is critical. In addition, knowledge about the expected model provided by additional priors can enhance or, with a wrong design, compromise the tracking. Sections 3.2 and 3.3 defined several priors and likelihoods for the AHHM. To recap, those adopted for the HPPF are:

The **global likelihood** is independent of the level and the partition and from equation (3.66) results in:

$$\lambda_l^{\Phi}\left(\mathrm{O}_t|\mathrm{p}_t(i)\right) = \lambda_G\left(\mathrm{O}_t|\mathrm{p}_t(i)\right) \tag{5.40}$$

The **local likelihood**, with the expression from equation (3.52) is:

$$\lambda_l^{\Phi}\left(\mathrm{O}_t|\mathrm{p}_t(i)\right) = \prod_{j=1}^{c}\prod_{\phi\in\Phi} \lambda_e^{\alpha_e}(\mathrm{E}_t^j|\mathrm{p}_t^{\phi}(i))\lambda_s^{\alpha_s}(\mathrm{S}_t^j|\mathrm{p}_t^{\phi}(i))\lambda_c^{\alpha_c}(\mathrm{I}_t^j|\mathrm{p}_t^{\phi}(i)). \tag{5.41}$$

The weighting **prior**, for local levels only, is composed by the range prior, $\pi_r$, equation (3.20), and the maximum visibility prior, equation (3.32):

$$\pi_l^{\Phi}(\mathrm{p}_t(i)) = \prod_{\phi \in \Phi} \left[ \pi_r(\mathrm{p}_t^{\phi}(i)) \cdot \prod_{j=1}^{c} [\pi_v^j(\mathrm{p}_t^{\phi}(i))]^{\alpha_v} \right]. \tag{5.42}$$

The first limits the parameters around their anatomical range, while the second maximises the visibility of each body part in all the views.

Above, the three likelihoods and the visibility prior are individually enabled ($\alpha_x = 1$) or disabled ($\alpha_x = 0$), for $x \in \{e, s, c, v\}$.

Further, the colour likelihood requires a colour model, which is initialised in the first frame and updated frame by frame with equation (3.61), when $\alpha_u = 0.6$ or not, when $\alpha_u = 0$.

Various combinations of the above weighting components are tested in table 5.12, comparing the tracking efficiency to the global likelihood. Edge, colour and silhouette likelihoods are individually switched on, then combinations with and without size and colour updates are tested.

| Likelihood | Global | $\alpha_s$ | $\alpha_e$ | $\alpha_c$ | $\alpha_v$ | Update | 3D error [mm] | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | rel | abs |
| Global S & E | Y | – | – | – | – | – | 109.5 | 111.9 |
| S | N | 1 | 0 | 0 | 0 | – | 105.0 | 107.7 |
| E | N | 0 | 1 | 0 | 0 | – | 165.9 | 322.6 |
| C | N | 0 | 0 | 1 | 0 | – | 173.2 | 178.8 |
| S & E | N | 1 | 1 | 0 | 0 | – | 112.7 | 115.5 |
| S, E & C | N | 1 | 1 | 1 | 0 | Y | 105.5 | 102.9 |
| S, E, C & V | N | 1 | 1 | 1 | 1 | Y | 86.4 | 88.1 |
| S, E, C & V, NCMU | N | 1 | 1 | 1 | 1 | N | 92.8 | 91.1 |

Table 5.12: Tracking error dependence on the likelihood function on supplementary priors. 3D and 2D (camera C1 and C2) relative and absolute errors. S – silhouette, E – edge, C – colour, V – visibility, NCMU – no colour model update.

From table 5.12 several conclusions are drawn:

- the local silhouette and edge likelihoods are comparable with the corresponding global likelihood, but are better than the individual local edge or colour likelihoods.
- the best standalone likelihood is the silhouette, confirming the results of Balan *et al.* [141],

- the addition of edge and colour likelihoods does not improve the tracking error compared to the silhouette
- the visibility prior provides additional tracking accuracy,
- the likelihood is better if the colour model is updated over the time.

Improvements were expected for the local compared to the global likelihoods and with the edge measurements. Noisy observations (*i.e.* edge detection) and bad initialisation of the colour model are reasons for the missing improvements. Silhouettes of moving objects are robustly extracted, therefore prove to be the best observations.

### 5.4.5 Stochastic constants

The particle generation depends strongly on the `GetNextBFV` function (algorithm 5). This function has the stochastic constants $\sigma_P$, $\sigma_L$, $\sigma_S$ and $\sigma_N$, in addition to the learnt MCM model. If a constant is small then it constrains the prediction close to the mean of the cluster, while if the constant is large then the prediction is far, and unrelated to the MC. Table 5.13 shows the tracking error variation around their found optimal values. That are for **pose**, **limbs** and **speed** modes $\sigma_P = \sigma_L = \sigma_S = 1$, and for the **normal** mode $\sigma_N = 0$.

A $\sigma_X$ equal to zero implies that the mean pose to which the model changes has no stochastic component. For pose, limb and Gaussian drift-based transitions, it can be observed that a zero or near zero value results in a low error, after which there is a transitory peak before another minimum. This suggests that without stochastic pose-components, mean transitions with random selection of the propagation type and modes are sub-optimum solutions, saving the overhead from random normally distributed sample generation when fast processing is critical.

### 5.4.6 Particle survival

In a high dimensional space the number of required particles for a good density estimate increases exponentially [117,128]. While particle set degeneration [60] is inconvenient, only viable particles must propagate to the next level, while preserving the multi-modality and variety.

In order to focus particles towards the peaks of the distribution, badly characterised by the likelihood function (*e.g.* due to depth ambiguity, occlusions, low contrast) in line

| $\sigma_P$ | $\sigma_L$ | $\sigma_S$ | $\sigma_N$ | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | rel | abs | rel | abs | rel | abs |
| 0.00 | 1.00 | 1.00 | 0.00 | 105.1 | 106.1 | 20.2 | 21.6 | 18.7 | 19.0 |
| 0.25 | | | | 116.5 | 116.5 | 21.4 | 22.4 | 20.6 | 20.7 |
| 0.50 | | | | 85.4 | 89.8 | 17.0 | 18.9 | 16.9 | 17.7 |
| 0.75 | | | | 89.8 | 92.4 | 17.4 | 18.9 | 17.9 | 18.3 |
| 1.00 | | | | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| 1.25 | | | | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| 1.00 | 0.00 | 1.00 | 0.00 | 87.0 | 89.1 | 17.1 | 18.5 | 17.2 | 17.6 |
| | 0.25 | | | 112.8 | 111.0 | 19.8 | 20.8 | 19.9 | 19.8 |
| | 0.50 | | | 101.2 | 104.1 | 18.6 | 20.1 | 19.0 | 19.6 |
| | 0.75 | | | 104.6 | 105.3 | 18.9 | 20.3 | 19.3 | 19.5 |
| | 1.00 | | | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| | 1.25 | | | 93.3 | 94.3 | 17.6 | 19.3 | 18.3 | 18.5 |
| 1.00 | 1.00 | 0.00 | 0.00 | 111.4 | 112.3 | 20.6 | 22.1 | 19.6 | 19.9 |
| | | 0.25 | | 92.7 | 93.2 | 17.6 | 19.1 | 17.8 | 18.2 |
| | | 0.50 | | 102.9 | 101.6 | 19.2 | 20.3 | 19.0 | 19.2 |
| | | 0.75 | | 97.9 | 100.8 | 18.6 | 20.3 | 18.2 | 18.6 |
| | | 1.00 | | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| | | 1.25 | | 89.5 | 90.2 | 17.3 | 18.6 | 17.6 | 17.9 |
| 1.00 | 1.00 | 1.00 | 0.00 | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| | | | 0.05 | 102.6 | 101.1 | 18.8 | 19.9 | 18.8 | 18.9 |
| | | | 0.10 | 91.3 | 93.2 | 17.8 | 19.4 | 17.8 | 18.0 |
| | | | 0.20 | 95.9 | 94.4 | 18.1 | 19.3 | 18.3 | 18.4 |
| | | | 0.40 | 89.1 | 91.6 | 17.5 | 19.3 | 17.5 | 18.0 |
| | | | 0.80 | 91.7 | 94.1 | 17.5 | 19.1 | 17.8 | 18.5 |

Table 5.13: Tracking error dependence on propagation stochastic constants. Each constant is varied around their optimal point of all four.

14 of the HPPF (algorithm 10) function, `EnhanceCloud` applies two independent methods to emphasise weights with higher values.

First, the *weight scaling* stretches the range of the weights to $[0, 1]$ by the scaling

$$\bar{w}_t^{\Phi}(i) = \frac{\tilde{w}_t^{\Phi}(i) - \min_i \tilde{w}_t^{\Phi}(i)}{\max_i \tilde{w}_t^{\Phi}(i) - \min_i \tilde{w}_t^{\Phi}(i)} \tag{5.43}$$

Second, elimination finds the highest $\tau$ percent of weights and discards all lower value particles:

$$T = \breve{w}_t^{\Phi}(\tau \cdot n_p), \tag{5.44}$$

where $\breve{Maw}_t^{\Phi}$ is the increasingly ordered sequence of weights $\bar{\mathrm{w}}_t^{\Phi}$. Finally, all particles with a weight lower then the threshold $T$ are removed:

$$\check{w}_t^{\Phi}(i) = \begin{cases} \bar{w}_t^{\Phi}(i), & \text{if} \quad \bar{w}_t^{\Phi}(i) \geq T \\ 0 & \text{otherwise} \end{cases} \tag{5.45}$$

In the performed tests, the lower $\tau = 80\%$ of particles were discarded.

Both enhancement steps can be disabled by $\bar{w}_t^{\Phi}(i) = \tilde{w}_t^{\Phi}(i)$ and respectively $\check{w}_t^{\Phi}(i) = \bar{w}_t^{\Phi}(i)$. Table 5.14 compares the effect of the two against the tracking with no scaling or elimination.

| Weight scaling | Elimination | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|---|
| | | rel | abs | rel | abs | rel | abs |
| no | no | 107.6 | 107.1 | 19.3 | 20.7 | 19.5 | 19.7 |
| no | yes | 118.4 | 119.4 | 20.6 | 22.1 | 21.0 | 21.1 |
| yes | no | 103.7 | 100.8 | 19.3 | 20.3 | 18.9 | 18.8 |
| yes | yes | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |

Table 5.14: Tracking result for fitness survival.

The *weight stretching* emphasises small differences of particles and therefore reduces the tracking error. Combined with the *elimination*, it provides the lowest errors, although the *elimination* alone is not effective, possibly because reducing the particle variety inhibits the multiple hypotheses of the PF and therefore recovery from incorrect hypotheses.

### 5.4.7 Optimised parameters

To conclude the parameter analysis, the optimal parameters of the HPPF-MCM tracking on the HumanEva dataset are summarised in table 5.15.

The generality of these parameters to track other video sequence will be shown in section 5.6.

$n_{\mathcal{C}}$ and $l_m$ differ from the optimal values for behaviour recognition, obtained in section 4.6.6. This suggest that further analysis on the effect of $n_{\mathcal{C}}$ and $l_m$ is required for the full tracking and behaviour analysis framework. This will be discussed in chapter 6.

## 5.5 Multiple vs. single camera tracking

Ideally, with good motion prediction and powerful likelihoods, a single or limited camera views provide good tracking. In practice, this is not the case, for the following reasons:

| Parameter | Value |
|---|---|
| $n_p$ | 150 |
| $n_{\mathcal{C}}$ | 100 |
| $l_m$ | 3 |
| $\alpha_s$ | 1 |
| $\alpha_e$ | 1 |
| $\alpha_c$ | 1 |
| $\alpha_v$ | 1 |
| $\alpha_u$ | 0.6 |
| $\sigma_P$ | 1 |
| $\sigma_L$ | 1 |
| $\sigma_S$ | 1 |
| $\sigma_N$ | 0 |
| weight scaling | yes |
| elimination | yes |
| tracking estimate | windowed-mean |

Table 5.15: Optimised parameters of the HPPF-MCM tracker.

- perspective ambiguity: a single image does not contain sufficient information for 3D reconstruction,
- self occlusion: body parts are frequently obscured by other body parts,
- occlusion by scene objects: temporary obstructions of large parts of the human,
- occlusion by other moving objects: adds uncertainty about the location of the obscured part

The effects of the number of cameras on tracking reliability are shown in table 5.16. With multiple cameras (two or three), the relative and absolute errors are both better than the enhanced PF-SIR filter. Since the relative and absolute errors are close, the pose estimate is good. However, when three cameras are reduced to two, the absolute error is increased by 47%. Whether this is acceptable or not depends on the application of the tracking. With a single camera, high values of the absolute 3D errors suggest that the 3D positions are poorly recovered, since depth is obtained from the model scale variations only.

## 5.6 Tracking results

Tracking results of the HPPF-MCM with the optimised parameters from table 5.15 on the HumanEva, CAVIAR and i-LIDS datasets are presented next.

| Cameras | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|
| | rel | abs | rel | abs | rel | abs |
| HPPF-MCM: C1, C2, C3 | 86.4 | 88.1 | 16.8 | 18.3 | 17.3 | 17.5 |
| HPPF-MCM: C1, C2 | 124.2 | 129.5 | 21.7 | 22.6 | 21.2 | 21.2 |
| HPPF-MCM: C1 | 192.1 | 834.4 | 39.2 | 44.4 | 28.6 | 137.5 |
| HPPF-MCM: C2 | 162.1 | 426.2 | 29.0 | 78.2 | 25.3 | 24.9 |
| PF-SIR | 146.0 | 152.5 | 25.2 | 27.2 | 23.6 | 24.1 |

Table 5.16: Tracking error for reduced camera input. HPPF-MCM with three to one camera views is compared to PF-SIR.

### 5.6.1 HumanEva-I test sequences

The tracking particularities of each sequence are important, in addition to the mean error for the seven test sequences (table 5.1). Table 5.17 shows that *S1 Gesture 1* and *S1 Walking 1* have the lowest errors. This was expected, since sequences of subject S1 were included in the training data, however not the same frames as used for testing. *Gesture* also has low error, since the activity is performed in an approximately constant position and only one hand motion is involved. Errors are in the range 7.4–12.5 cm, which is not accurate enough for demanding applications such as computer animation, but should be adequate for behaviour analysis. The higher errors of *Jog* and *S3 Walking 1* are expected to lower the behaviour recognition.

| Sequence | 3D error [mm] | | C1 2D error [pixels] | | C2 2D error [pixels] | |
|---|---|---|---|---|---|---|
| | rel | abs | rel | abs | rel | abs |
| *S1 Walking 1* | 70.3 | 70.5 | 14.4 | 14.1 | 15.7 | 15.3 |
| *S1 Jog 1* | 106.4 | 104.9 | 18.9 | 18.7 | 20.9 | 19.9 |
| *S1 Gesture 1* | 46.2 | 52.1 | 11.1 | 11.5 | 11.9 | 12.5 |
| *S3 Walking 1* | 101.4 | 105 | 18.4 | 19.6 | 16.9 | 17.2 |
| *S3 Jog 1* | 121.5 | 121.3 | 24.7 | 24.9 | 21.9 | 21.6 |
| *S3 Gesture 1* | 74.7 | 75.8 | 14.8 | 14.6 | 16.3 | 16.0 |
| *S2 Throw/Catch 1* | 84.2 | 86.9 | 15.7 | 24.7 | 17.4 | 19.8 |

Table 5.17: Tracking errors for seven HumanEva sequences.

### 5.6.2 HumanEva-II test sequences

In the interests of more objective evaluation, the authors of the HumanEvaII dataset withheld the ground truth for the HumanEvaII [194] dataset. The tracking results sub-

mitted for evaluation online[1], result in the 3D and 2D absolute errors for these sequences, presented in table 5.18.

| | 3D/2D error | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 3D[mm] | | | Camera 1[pixel] | | | Camera 2[pixel] | | |
| Sequence | Set 1 | Set 2 | Set 3 | Set 1 | Set 2 | Set 3 | Set 1 | Set 2 | Set 3 |
| *S2 Combo 1* | 167.8 | 155.6 | 173.2 | 27.88 | 25.34 | 29.42 | 22.56 | 23.45 | 25.94 |
| *S4 Combo 4* | 121.9 | 132.6 | 144.2 | 22.38 | 23.19 | 25.66 | 20.40 | 22.14 | 23.29 |
| *S1 Walking 1* | 89.76 | N/A | N/A | 15.28 | N/A | N/A | 16.95 | N/A | N/A |

Table 5.18: Absolute 3D and 2D tracking errors for the three HumanEva 2 test sequences.

Figures 5.15 and 5.16 show two arbitrarily chosen sequences from the *S1 Walking 1* test frames. Both sequences (frames 9-33 and 219–267) are tracked reasonably well. For visualisation purposes, only every second frame of the later sequence is shown (*i.e.* the frame index increases by 6). The tracking quality over the whole sequence is good, though as the first four 3D error peaks in figure 5.24(a) show, the right whole or lower leg is temporarily lost, but recovered in the next semi-stride. Similarly, the last two error surges and the peaks around frames 210 and 297 are caused by lost lower leg tracks for periods of two to six frames. Considering their large DOF, the lower arms are severely affected for two 6–7 frame periods, at the two error peaks around frame numbers 240 and 345.

It is interesting to remark that raised 3D errors have corresponding high 2D error peaks for only one camera, suggesting that the respective camera provides weak measurements and is the reason for the lost 3D reconstruction.

Figures 5.17–5.21 show in all four camera views, and in the reconstructed 3D space, the *Walk* (frames 4–16), *Jog* (frames 461–475) and *Balance* (frames 766–781) segments of the *S2 Combo 1* sequence. For the whole sequence, the tracking error mainly results from the lower arms and lower legs: the first set of peaks up to about frame 800 are generated from temporarily lost lower legs after two legs overlap. The higher peak around frame number 184 accounts for swapped legs lasting for 3–4 frames.

While balancing, the image measurements are not powerful enough to support the wide inter-feet distance against the learnt poses, and therefore the more common standing pose is wrongly tracked for two out of four leg swings. The four error peaks at the end of diagrams from figure 5.24(b), correspond to the four balance poses. Of these, the middle two have correct leg poses and therefore lower errors, but mismatched arms are causing

[1] `http://vision.cs.brown.edu/humaneva/submit_results.html`

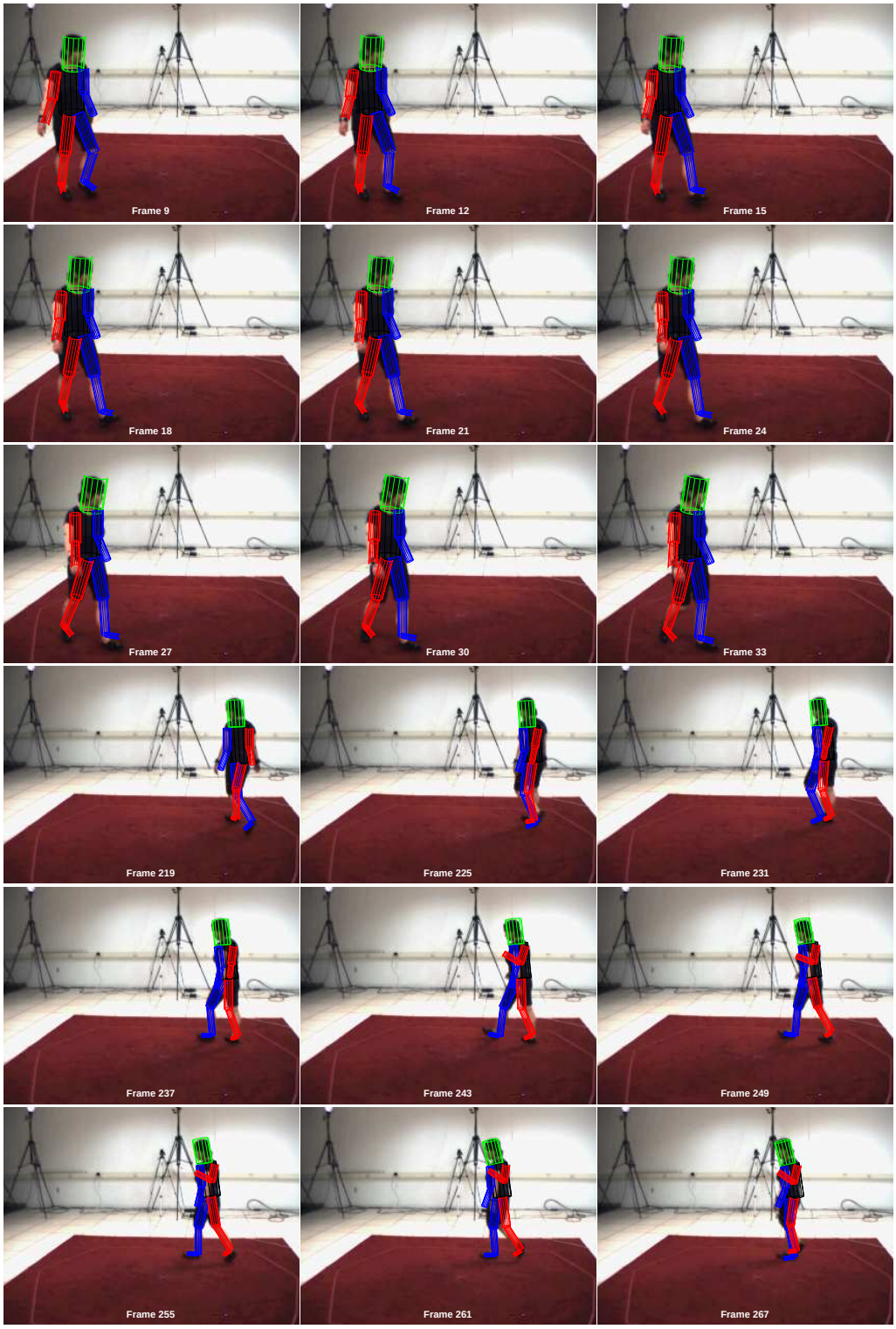



Figure 5.15: HumanEva *S1 Walking 1* camera C2 sequence: tracking results for frames 9–33 and 219–267 are superimposed with the input image [◊].

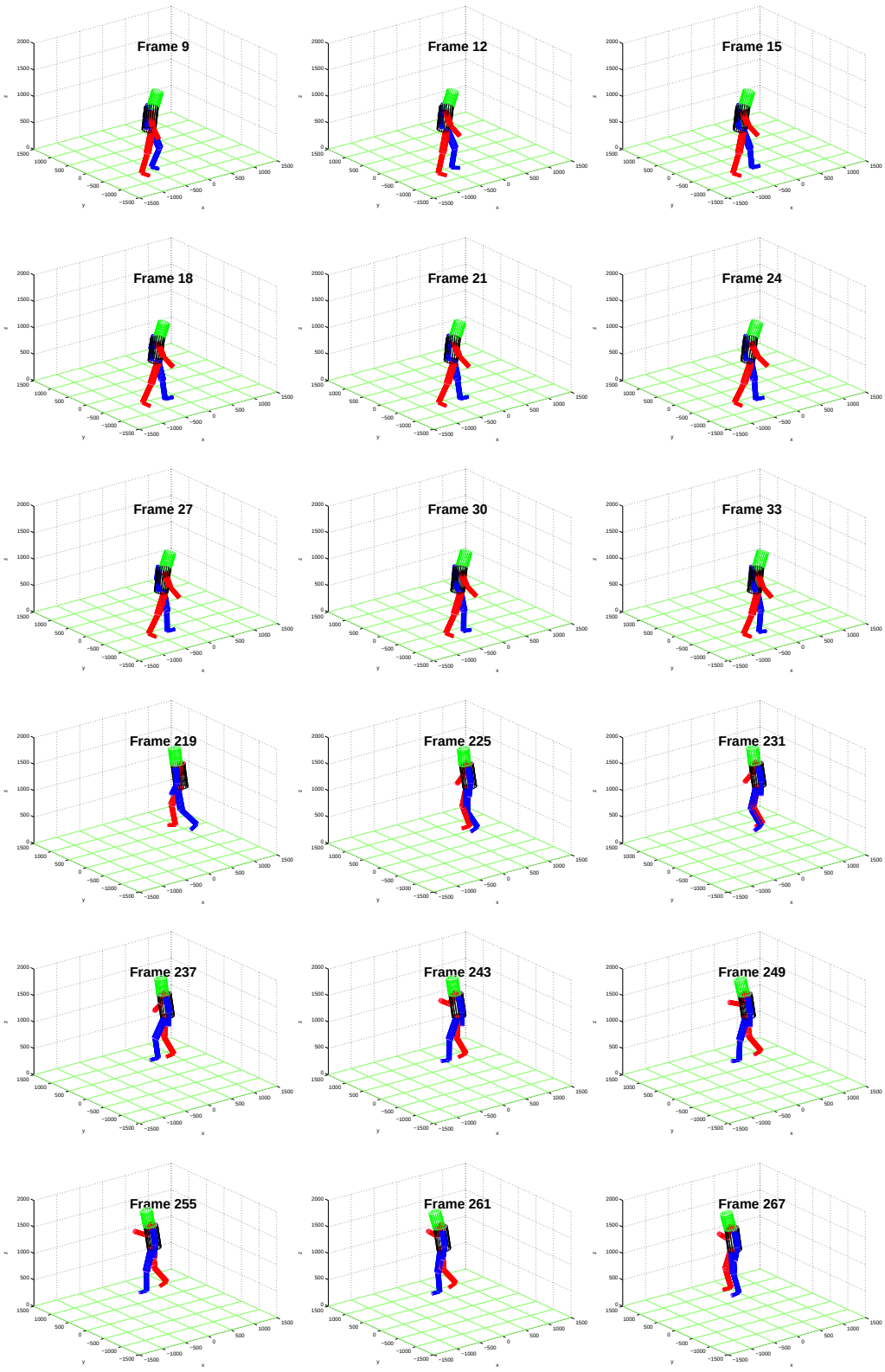



Figure 5.16: HumanEva *S1 Walking 1* 3D reconstruction: tracking results for frames 9–33 and 219–267 are visualised with the 3D model [◊].

the peaks.

Figures 5.22 and 5.23 capture the switch from jog to balance actions in the *S4 Combo 4* sequence. The sequence shows accurate tracking even without any training data for subject S4 or corresponding to balancing. The tracking over the whole sequence fails for only 2–3 frames, with inaccurate limb tracking, mainly of the lower arms, and with wrong arm tracking during the balancing activity. Figure 5.24(b) shows errors slightly over 100mm, with peaks around 200mm. The high error after frame number 302 is not observable on any of the 2D reprojections or the 3D model. This might been caused by the incorrect ground truth, however since this is not available, further investigations are impossible.

In general, the general position and orientation of the human body is well tracked during all sequences. For the *Walk* segment, the legs are tracked while arms are lost for two short durations. Tracking fails for the lower arms in both the *Combo* sequences, but it recovers with better observations. In the *Combo* sequences, both walking and jogging leg motions are well recovered, and even during jogging the slight centrifugal inclination towards the centre of the walking area is also visible in the 3D reconstructions. Some balancing activities have errors. The errors of the unseen S4 *Combo* sequence are comparable to the sequences of the trained subjects, suggesting a similar performance for other unseen subjects or activities.

The results on HumanEva-I and HumanEva-II are better than [114], with errors from 100 to 600 cm, in particular around 150–200 cm. The errors of 35–60 mm in [141] and 31.36 mm in [147] are lower than with HPPF, however they use the exact model of the HumanEva, here the model is different.

Table 5.19 compares the tracking results of multiple tracking algorithms, all evaluated on the HumanEvaII dataset. A relevant difference between HPPF and these algorithms, is the tracked model. All algorithms use the human model provided with the HumanEva, while HPPF uses the AHHM (chapter 3) developed before the dataset was published. The differences of the model generate an error of at least 23mm (see section 3.2.8, resulting in a handicap for evaluated HPPF.

Sigal *et al.* [137] and Howe [225] tracking errors for the *S1 Walking 1* sequence are higher then with the HPPF, and Sigal losses his target after 50 frames according to figure 5.25(b). The HPPF is comparable with Poppe's pose recovery [102], however his tracker performs also only with monocular cameras.

Tracking errors over the frames are compared in figures 5.25–5.27. All methods show

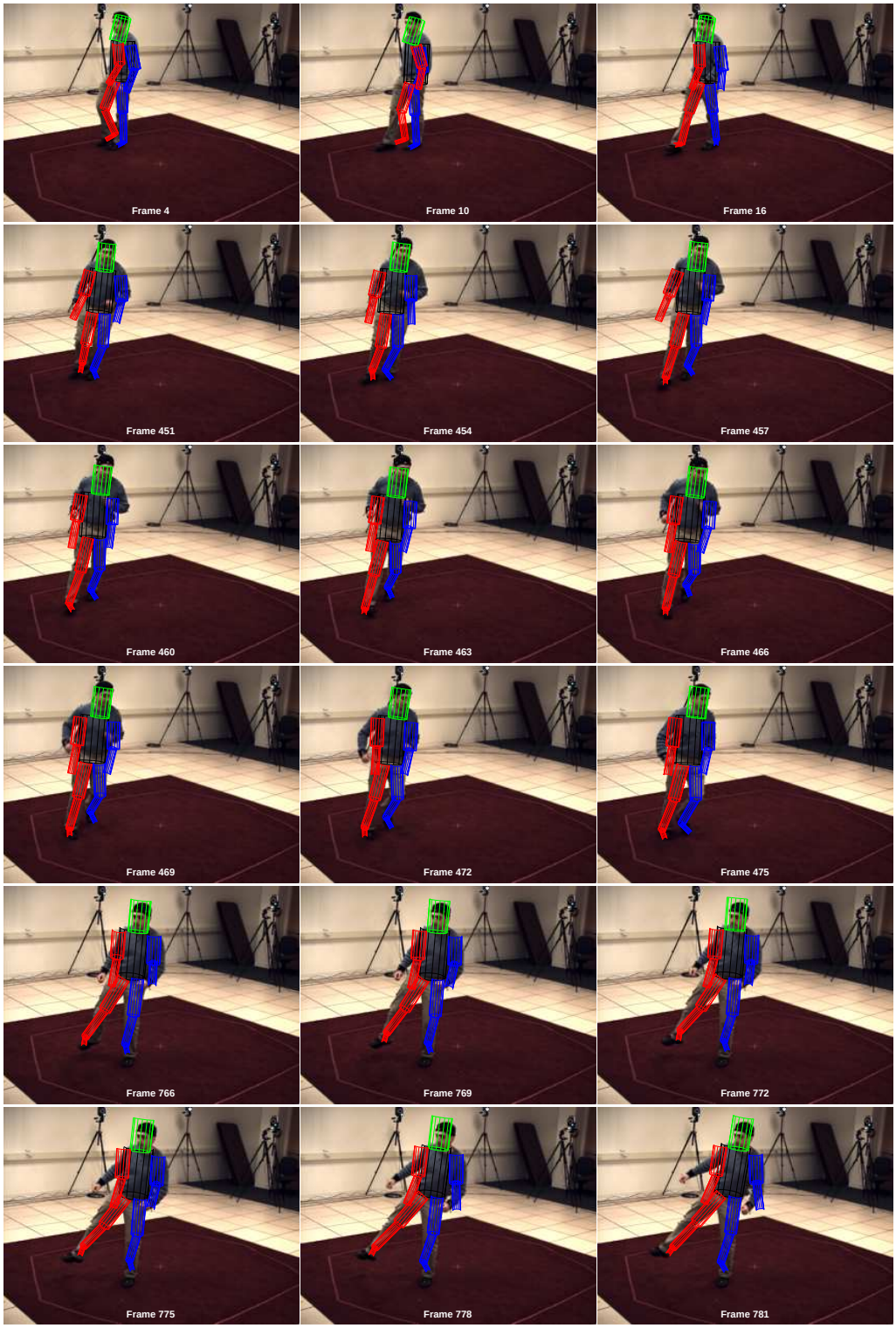



Figure 5.17: HumanEva *S2 Combo 1* camera C1 sequence: tracking results for frames of walking (frames 4–16), jogging (frames 461–475) and balance (frames 766–781) are superimposed with the input image [◊].

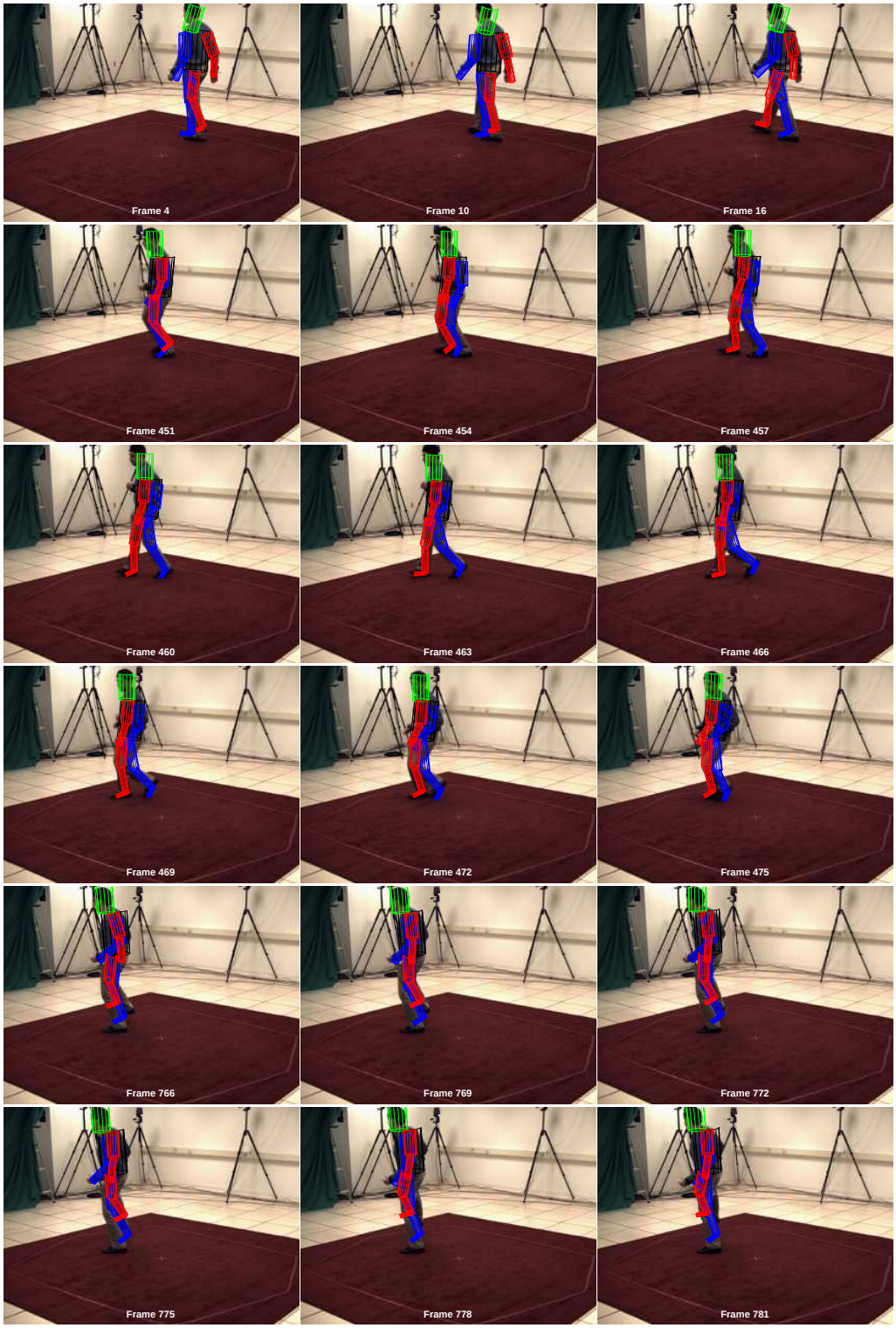



Figure 5.18: HumanEva *S2 Combo 1* camera C2 sequence: tracking results for frames of walking (frames 4–16), jogging (frames 461–475) and balance (frames 766–781) are superimposed with the input image [◊].

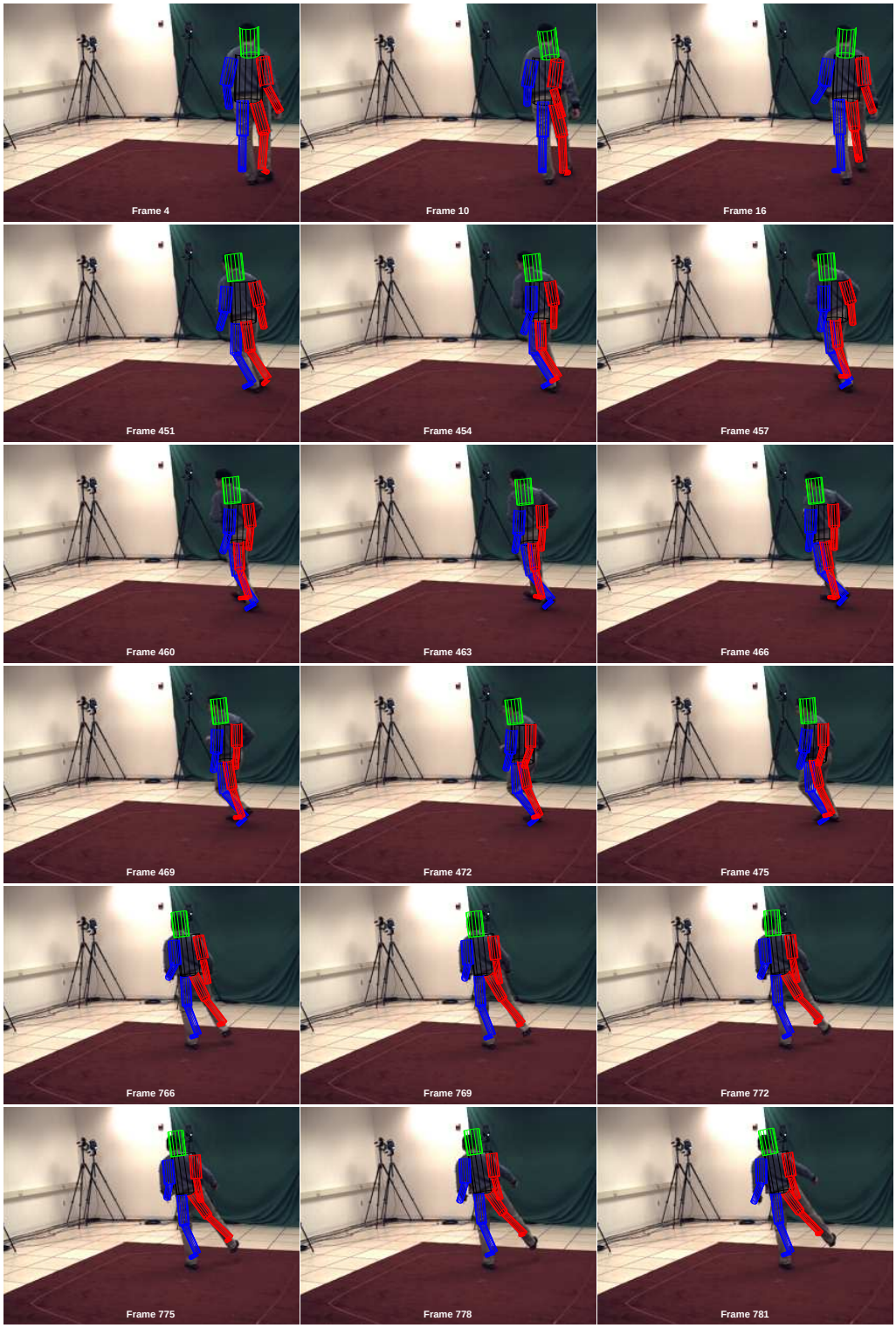



Figure 5.19: HumanEva *S2 Combo 1* camera C3 sequence: tracking results for frames of walking (frames 4–16), jogging (frames 461–475) and balance (frames 766–781) are superimposed with the input image [◊].

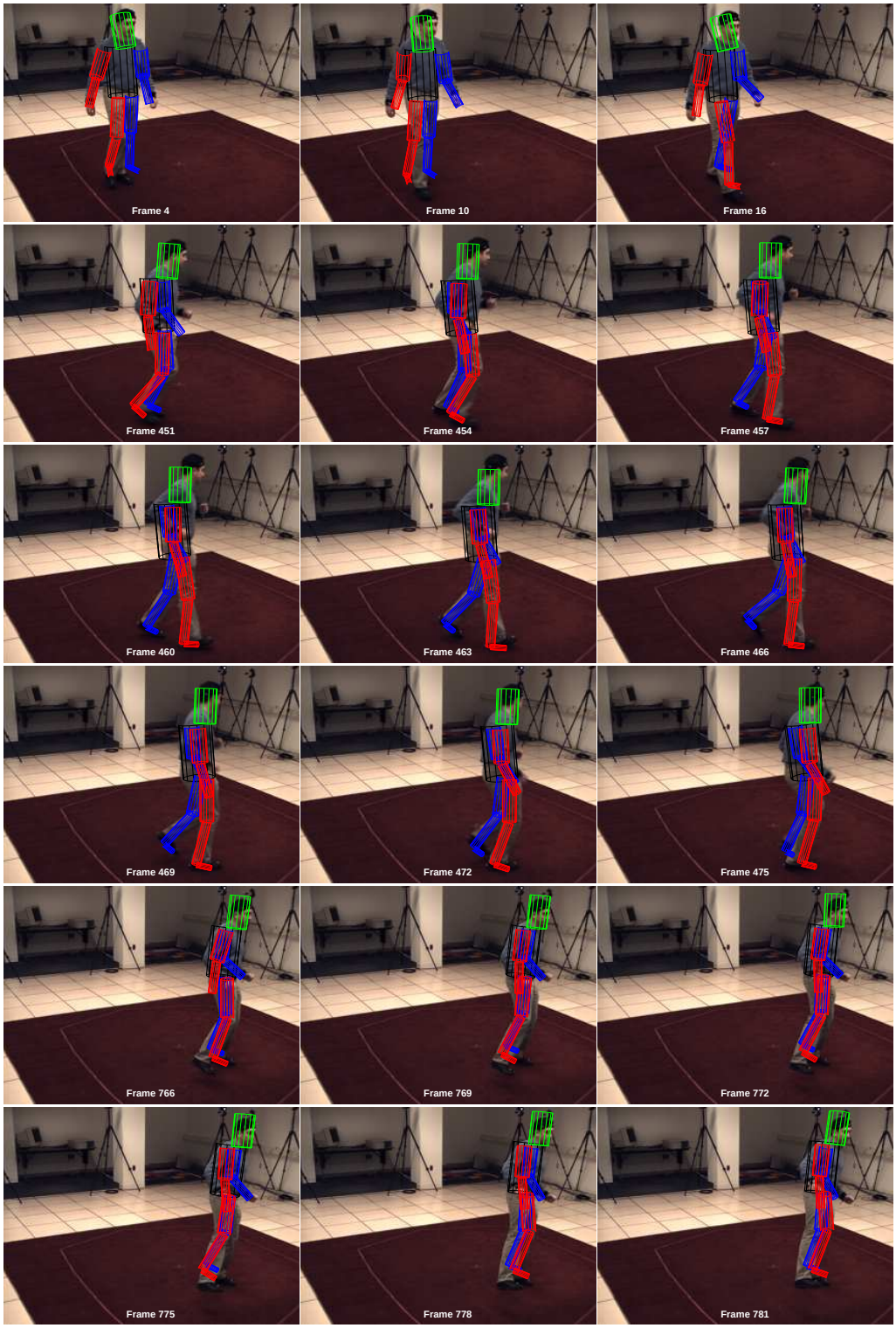



Figure 5.20: HumanEva *S2 Combo 1* camera C4 sequence: tracking results for frames of walking (frames 4–16), jogging (frames 461–475) and balance (frames 766–781) are superimposed with the input image [◊].

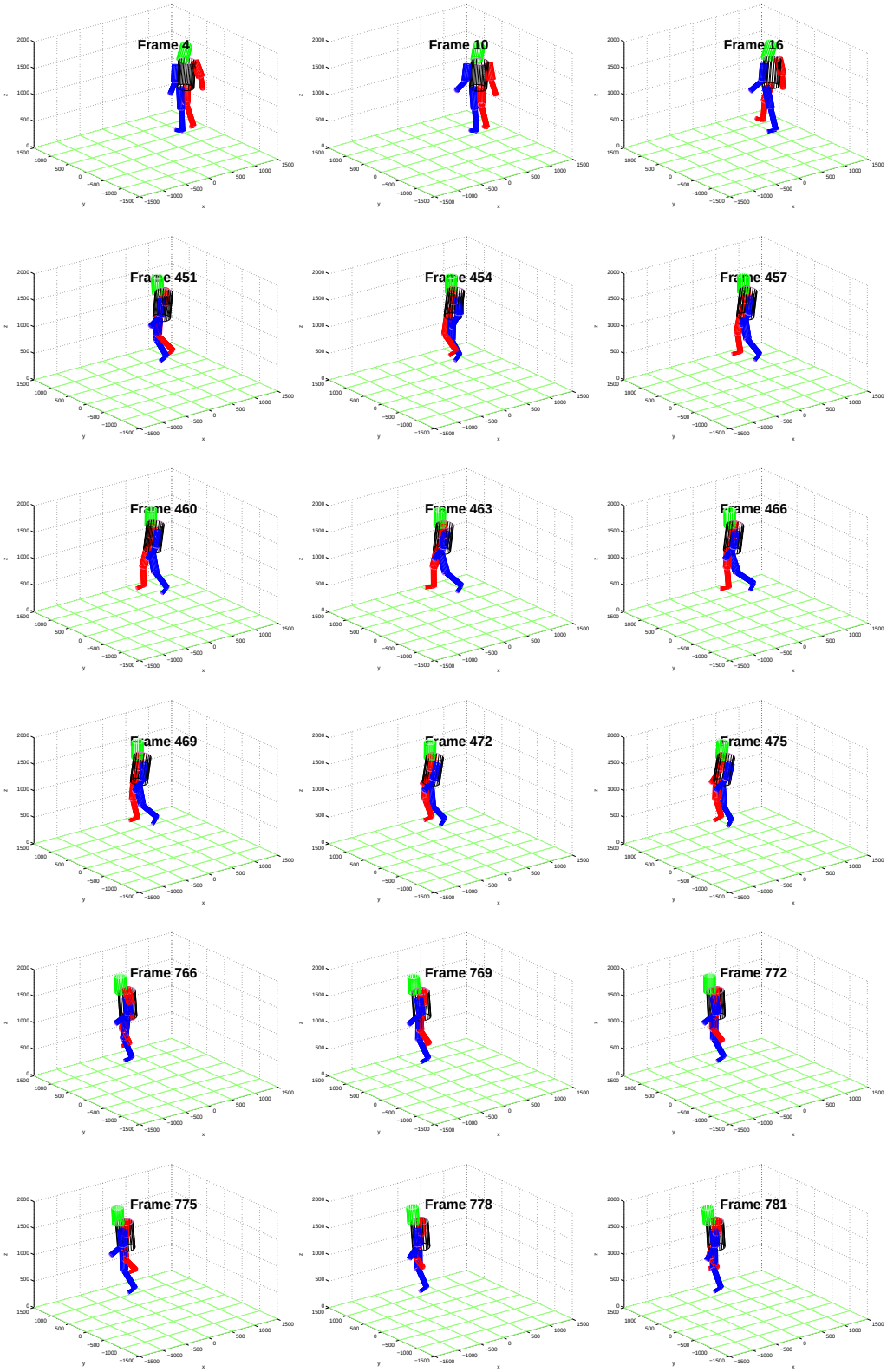



Figure 5.21: HumanEva *S2 Combo 1* 3D reconstruction: tracking results for frames of walking (frames 4–16), jogging (frames 461–475) and balance (frames 766–781) are visualised with the 3D model [◊].

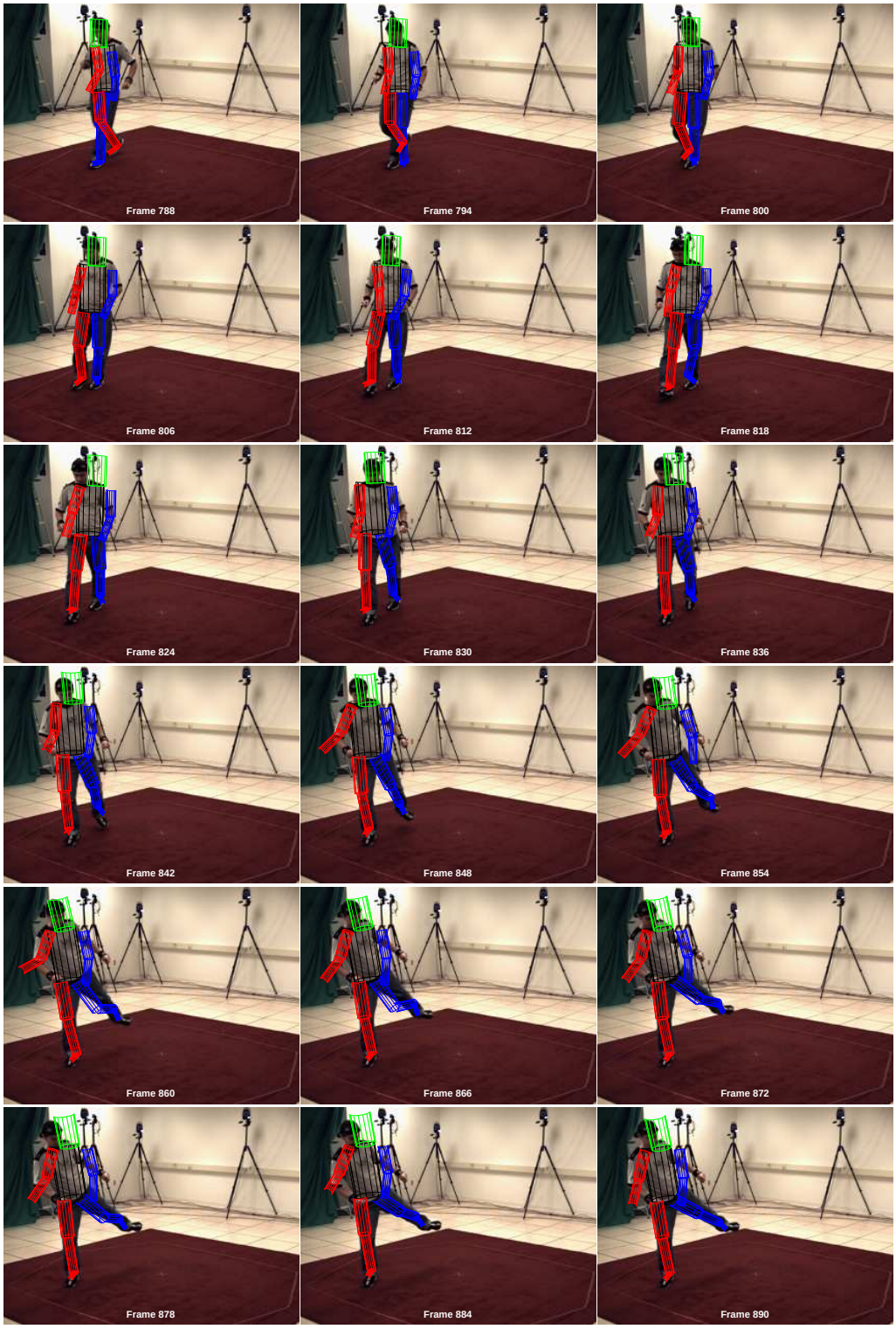



Figure 5.22: HumanEva *S4 Combo 4* camera C2 sequence I: tracking results for frames 788–890, the transition between walking and balancing [◊].

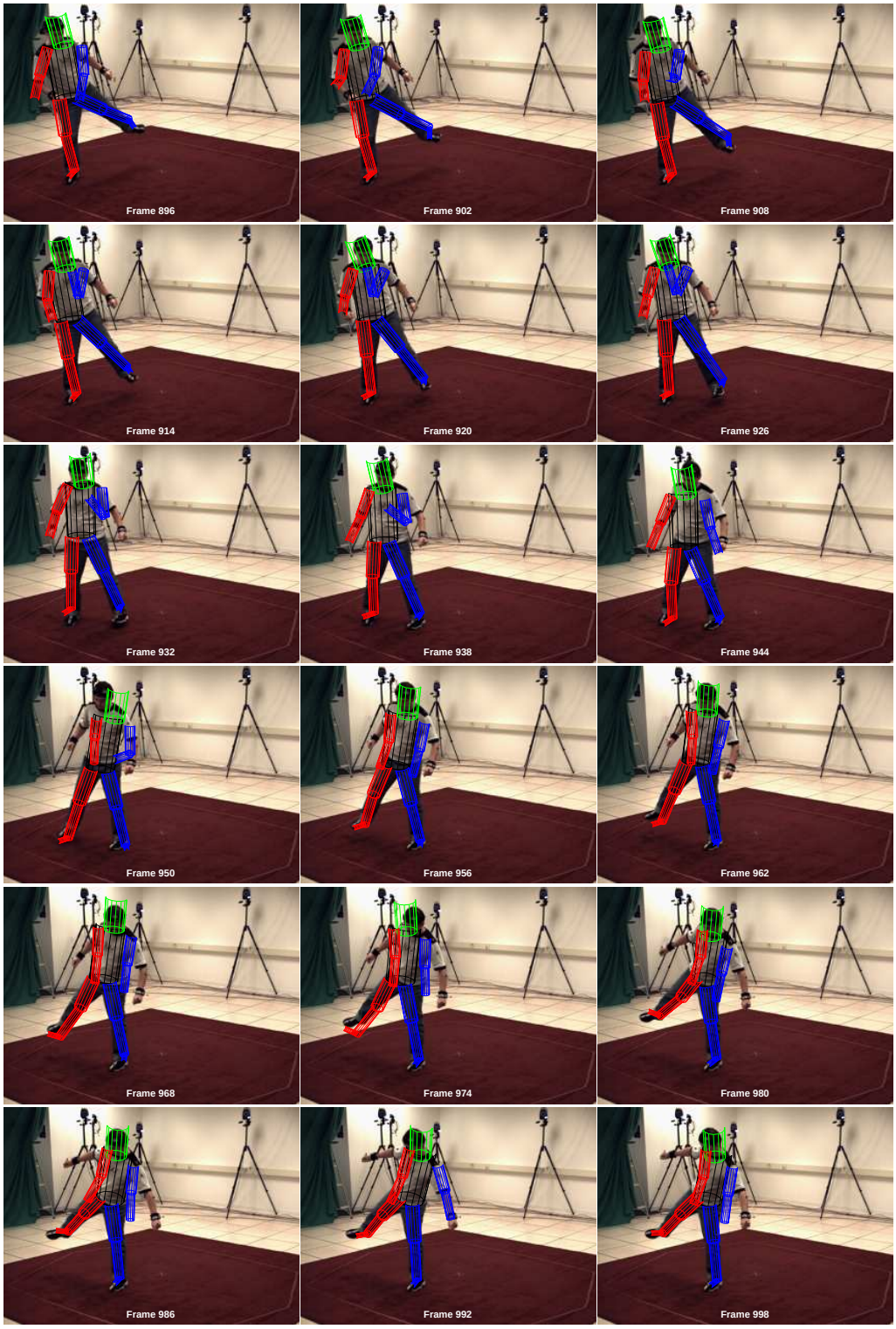
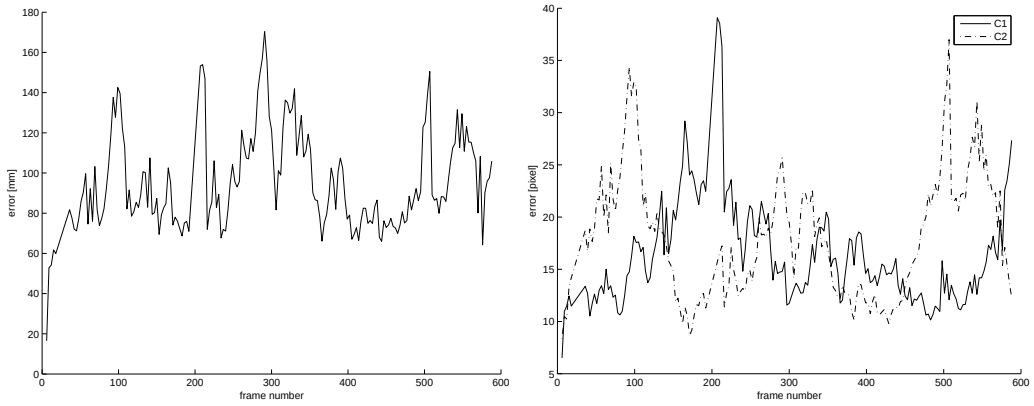
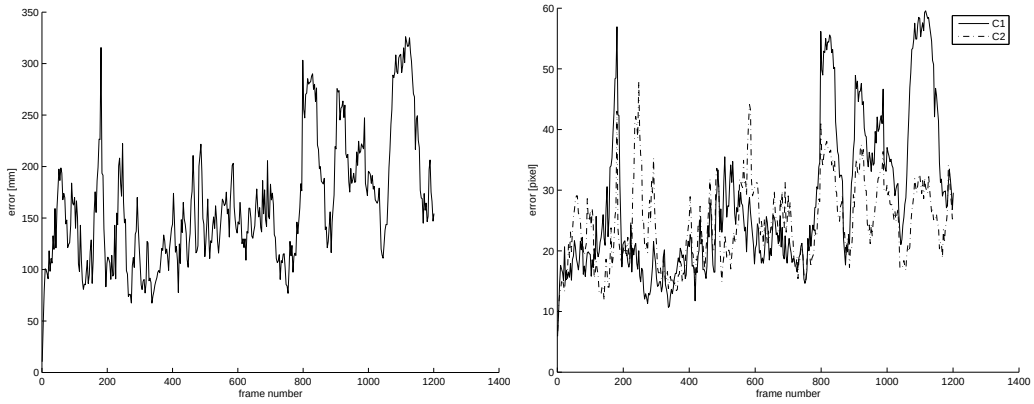


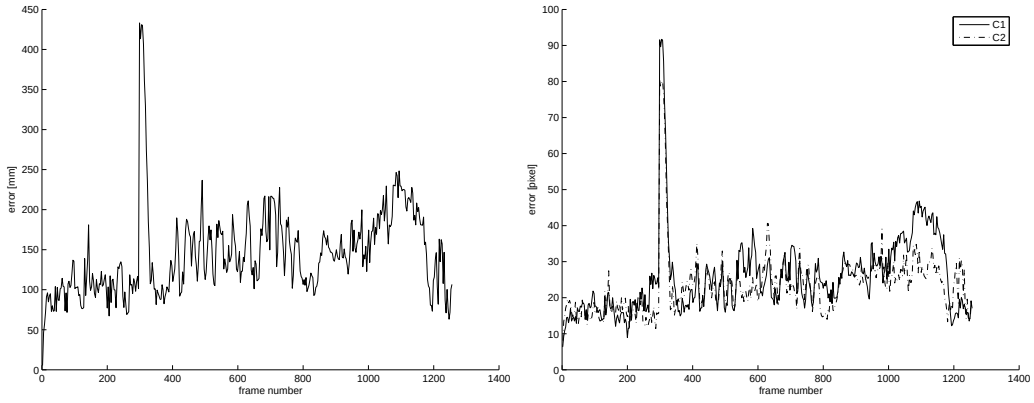

Figure 5.23: HumanEva *S4 Combo 4* camera C2 sequence II: tracking results for frames 896–998, the transition between walking and balancing [◊].




(a) *S1 Walking 1*




(b) *S2 Combo 1*




(c) *S4 Combo 4*

3D errors 2D errors

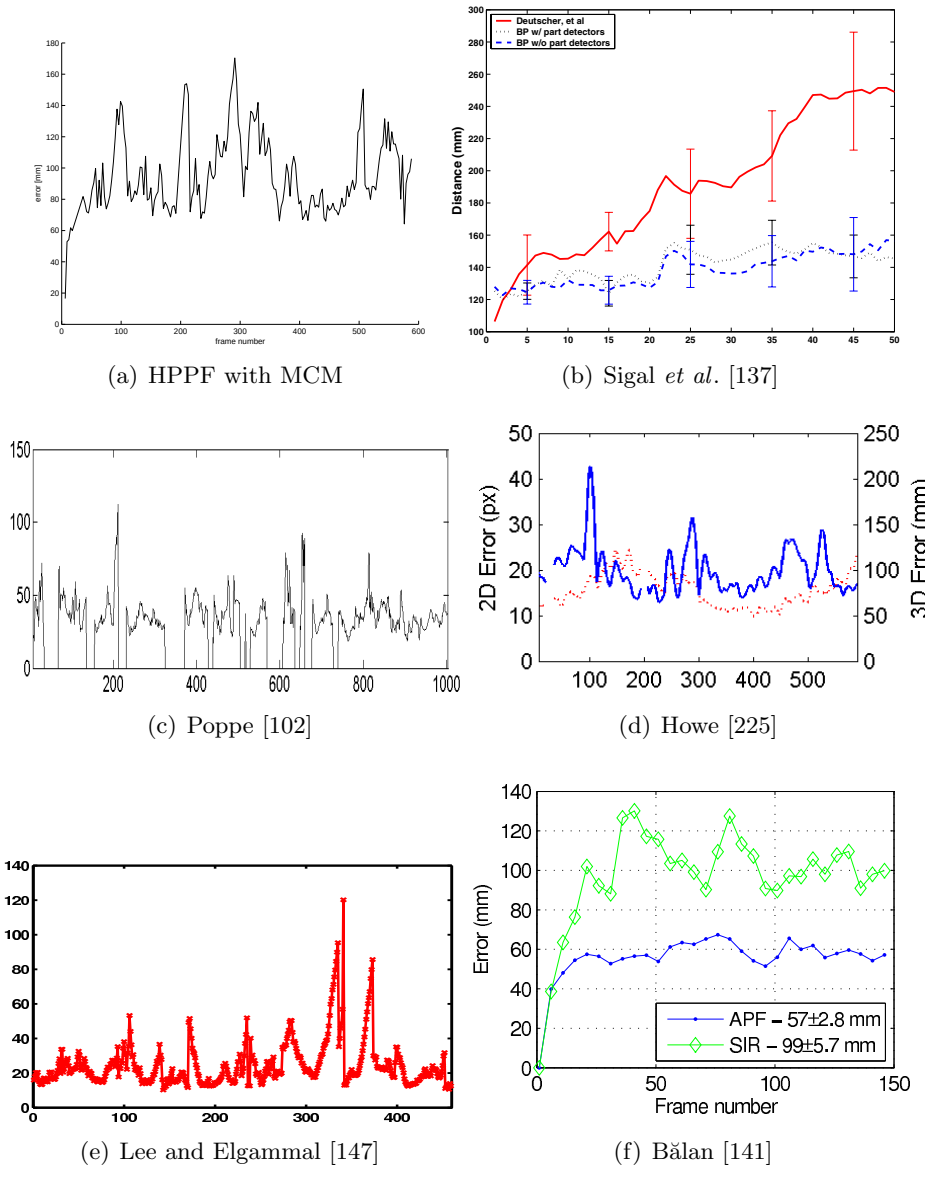
Figure 5.24: HumanEvaII 2D and 3D errors per frame for the three test sequences.

large error variations over the frame sequences. Reading figures 5.26(b) and 5.27(b) one can observer that Cheng's and Trivedi's error vary between 100mm and 400mm, suggesting a mean error higher than the reported mean errors between 92 to 210mm. It is interesting to remark the peek around frame 300, present in both figure 5.27(a), (b) and (d), pinpointing a possible incorrect MOCAP ground truth.

| Author | Sequence | Frames | 3D absolute [mm] | | |
|---|---|---|---|---|---|
| | | | Set 1 | Set 2 | Set 3 |
| Thesis | *S1 Walking 1* | Full | 89.76 | | |
| | *S2 Combo 1* | Full | 167.8 | 155.6 | 173.2 |
| | *S4 Combo 4* | Full | 121.9 | 132.6 | 144.2 |
| Sigal *et al.* [137] | *S1 Walking 1* | 50 | 140[b] | | |
| Lee and Elgammal [147] | *S1 Walking 1* | Full | 26.2[a] | | |
| Cheng and Trivedi [226] | *S2 Combo 1* | Full | 125[b] | 160[b] | 137[b] |
| | *S4 Combo 4* | Full | 92[b] | 210[b] | 177[b] |
| Howe [225] | *S1 Walking 1* | Full | 99[b] | | |
| | *S2 Combo 1* | *Walk* | 133[c] | | |
| | *S2 Combo 1* | *Combo* | 108[d] | | |
| | *S4 Combo 4* | *Walk* | 272[c] | | |
| | *S4 Combo 4* | *Combo* | 170[d] | | |
| Poppe [102] | *S1 Walking 1* | Full | 38[b] | | |
| | *S2 Combo 1* | Full | 109[bd] | 107[bd] | 170[bd] |
| | *S4 Combo 4* | Full | 145[bd] | 138[bd] | 179[bd] |
| Bălan *et al.* [141] | *S1 Walking 1* | 150 | 57 | | |
| | *S1 Walking 1* | 150 | 99 | | |

Table 5.19: Tracking errors compared with state of the art trackers evaluated on the HumanEvaII dataset. Notes: absolute errors are shown except when marked with [a] for relative errors; [b]estimate from figure; [c]mean from 3 camera views; [d]mean from 4 camera views

The HPPF, compared to [147], [225] and [102], has the drawback of requiring multiple cameras for stability, however it is not limited to specific actions and learnt silhouettes, but to a set of them that combine within the MCM to cover unseen situation. Table 5.20 shows that HPPF stands in the state of the art trackers, having a place in the multi-camera tracking. The tracking scenario is not limited to a single or limited number of activities (*i.e.* training with activities enhances the results but it is not mandatory) compared with other more accurate trackers.




(a) HPPF with MCM

(b) Sigal *et al.* [137]

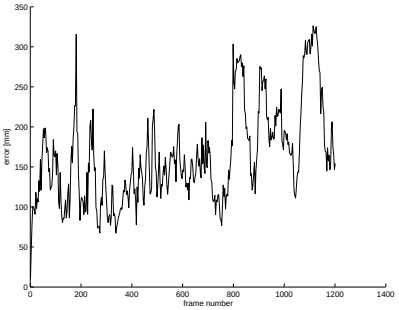
(c) Poppe [102]

(d) Howe [225]

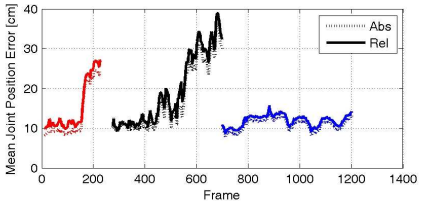
(e) Lee and Elgammal [147]

(f) Bălan [141]

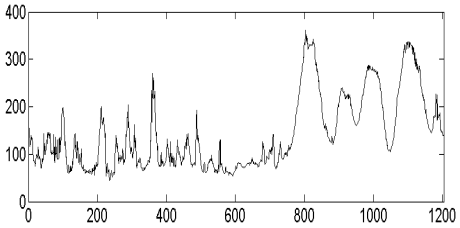
Figure 5.25: Tracking comparison on the *S1 Walking 1* sequence with figures reported by the authors: on horizontal the frame number, while on vertical the error in mm or in cm.



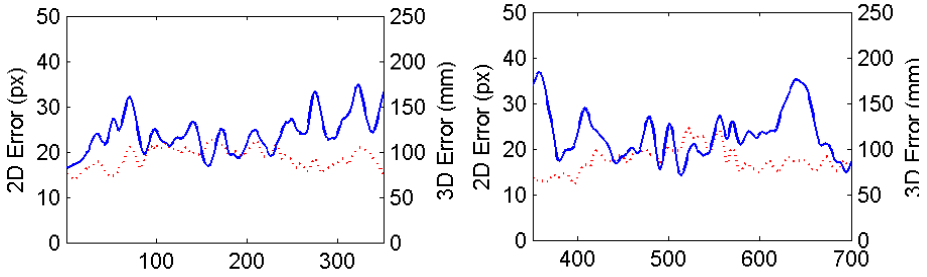

(a) HPPF with MCM




(b) Cheng and Trivedi [226]



(c) Poppe [102]

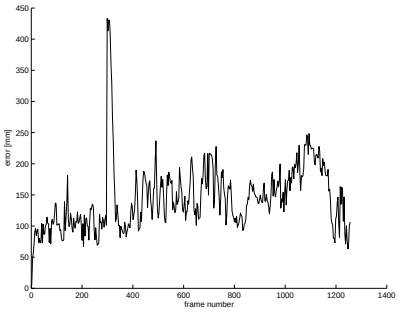



(d) Howe [225] Walk(left) and Combo(right)

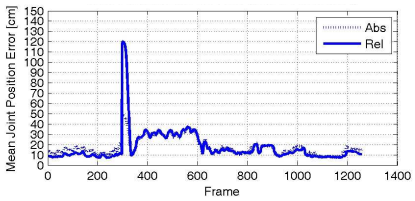
Figure 5.26: Tracking comparison on the *S2 Combo 1* sequence with figures reported by the authors: on horizontal the frame number, while on vertical the error in mm or in cm.

| Author | Method | Error | Cameras | Activity specific |
|---|---|---|---|---|
| Thesis | HPPF | | multi | no |
| Sigal *et al.* [137] | NBP | Lower | multi | no |
| Lee and Elgammal [147] | PF | Lower | mono | *Walk* only |
| Cheng and Trivedi [226] | Silhouette lookup | Comparable | multi | yes |
| Howe [225] | Silhouette and optical flow lookup | Comparable | mono | *Walk* and *Jog* only |
| Poppe [102] | Silhouette lookup | Comparable | mono | yes |
| Bălan *et al.* [141] | APF | Higher | multi | no |

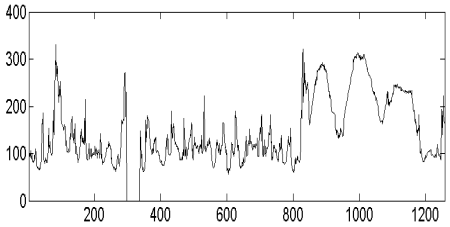
Table 5.20: Comparing HPPF with state of the art trackers: method, subjective error comparison, number of cameras and if method is‘restricted to the trained dataset.



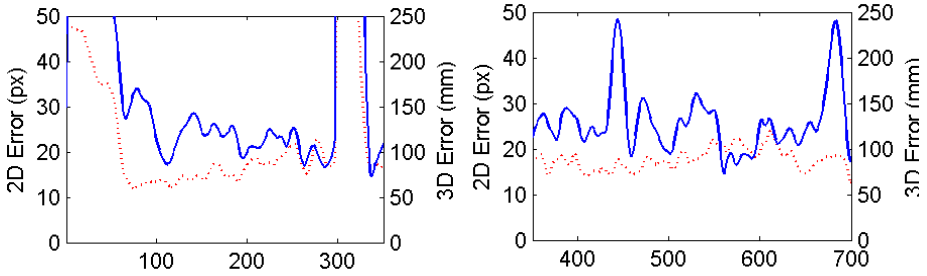

(a) HPPF with MCM




(b) Cheng and Trivedi [226]



(c) Poppe [102]




(d) Howe [225] Walk(left) and Combo(right)

Figure 5.27: Tracking comparison on the *S4 Combo 4* sequence with figures reported by the authors: on horizontal the frame number, while on vertical the error in mm or in cm.

### 5.6.3 Tracking CAVIAR sequences

CAVIAR sequences are harder to track for the following reasons: they have only two, wide baseline camera views with uncalibrated cameras; humans are smaller (30–150 pixels tall, compared to 220–440 pixels of the HumanEva) and present larger perspective scale variation (1:5 compared to 1:2); the observation field is larger; and multiple humans are present.

The camera is post-calibrated (section 3.1.4) and the tracked human position is manually initialised.

Since joint positions or 3D ground truth are not provided, evaluation is only visual. The tracking results of the *EnterExitCrossingPaths1* (figures 5.28–5.30) are similar to those for the HumanEva sequences. Tracking is generally good, with recovered details of body structure, although the same types of error occur temporarily for lower limbs.

In the *OneLeaveShopReenter1* sequence the subject walks then turns back. The HPPF-PPF tracked poses (figures 5.31–5.34), projected on the corridor view, suggest very good tracking, with exact recovery of the turn made while walking. The 3D reconstructions adequately represent the walking.

However, for both sequences, the 3D reconstruction presents an artefact; for several frames, the feet are below the ground plane, meaning that the height coordinate is inaccurate. Since the 2D re-projections are good, this suggests that only two camera views are not enough for *exact* articulated tracking.

### 5.6.4 iLIDS sequences

In contrast with the CAVIAR data, i-LIDS sequences have only single view images, and no calibration or ground truth. Again, manual initialisation and calibration (section 3.1.4) was applied.

The results from figures 5.35 and 5.36 show that pose tracking fails. There are multiple reasons for this: single camera, highly textured ground plane (edge likelihoods are compromised), inaccurate manual initialisation and calibration from single view. Even though the pose is not tracked, the human position is still successfully recovered.

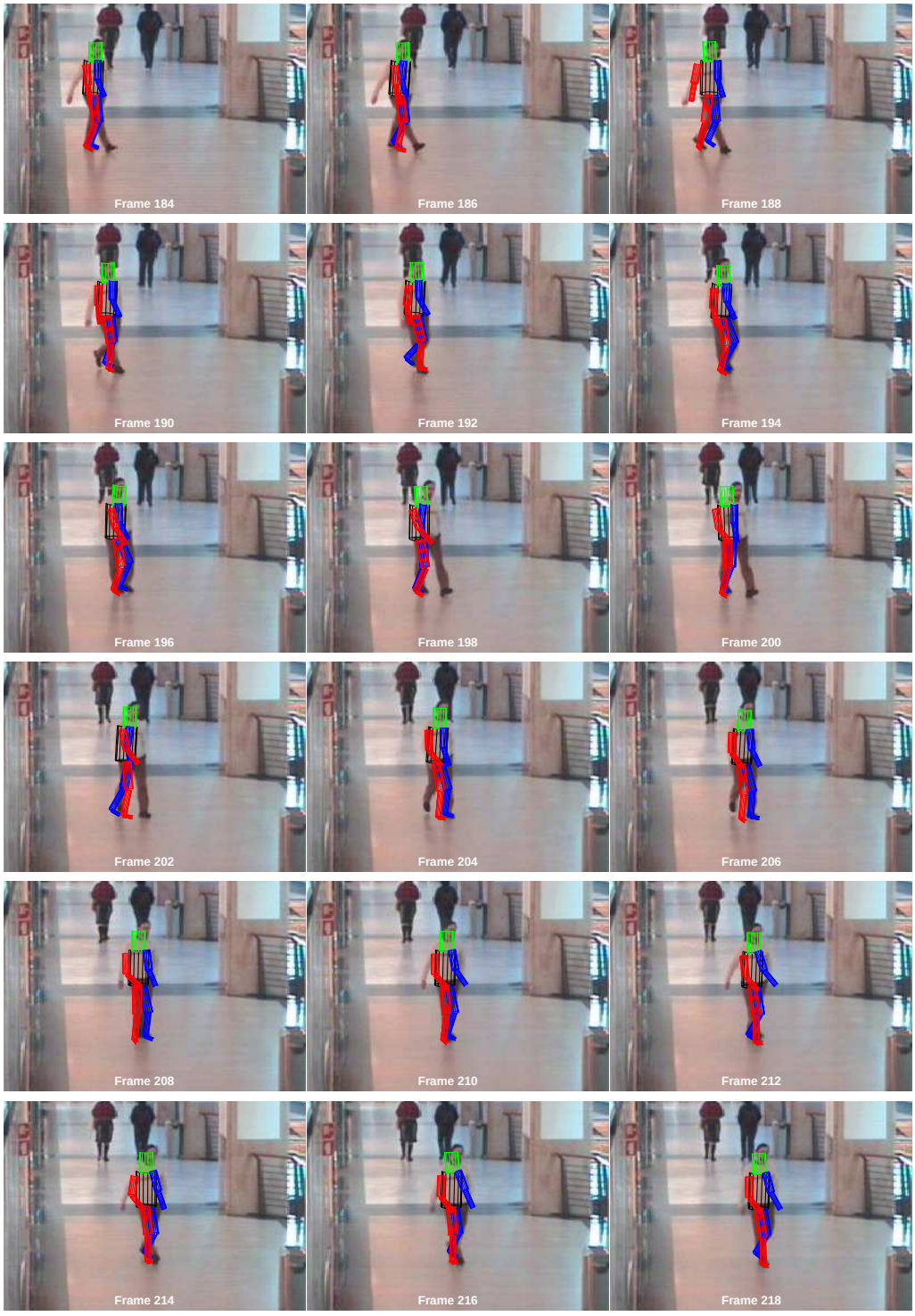



Figure 5.28: CAVIAR EnterExitCrossingPaths1 corridor sequence with every second frames in the range of 184–218 [◊].

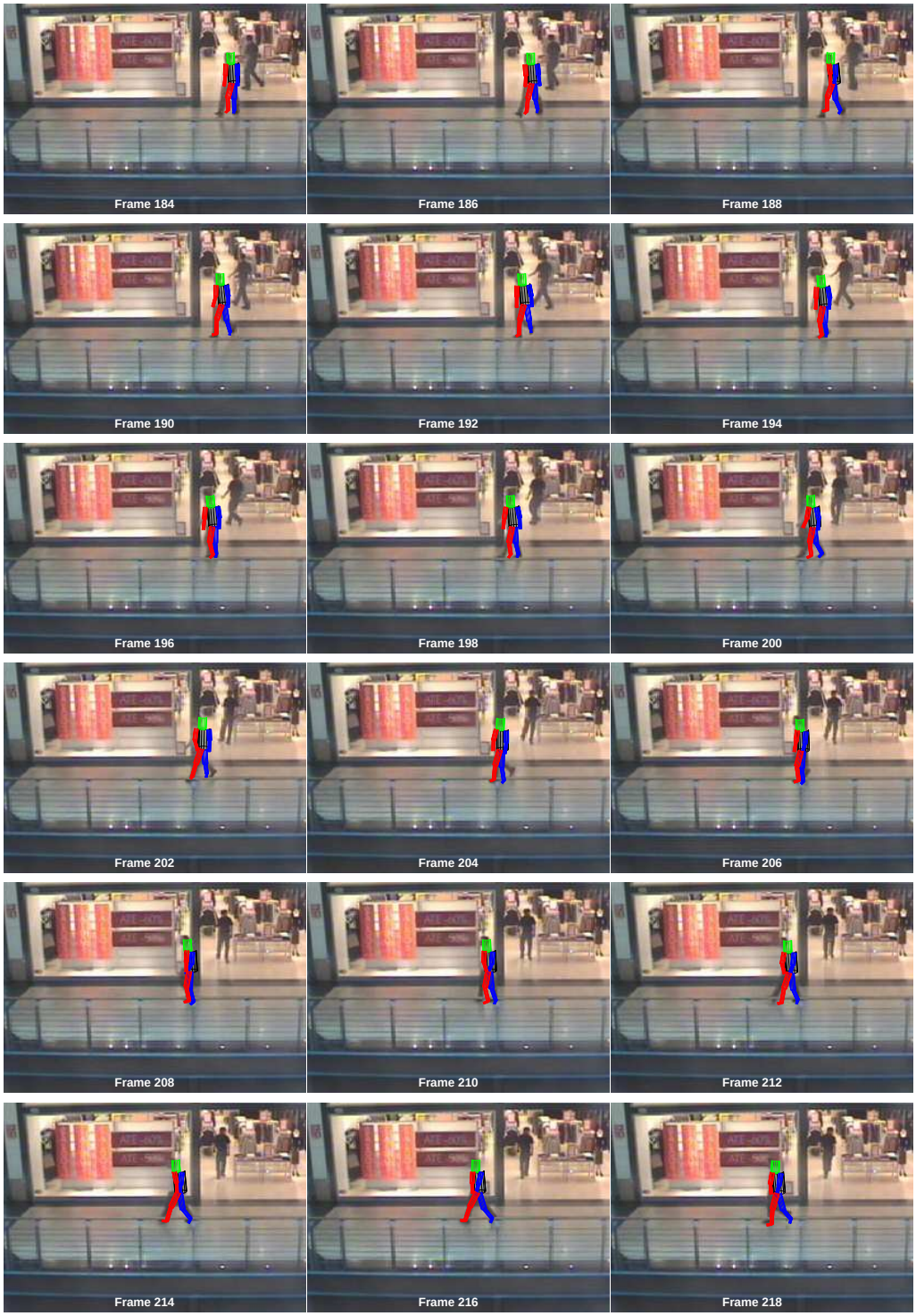



Figure 5.29: CAVIAR EnterExitCrossingPaths1 frontal sequence with every second frames in the range of 184–218 [◊].

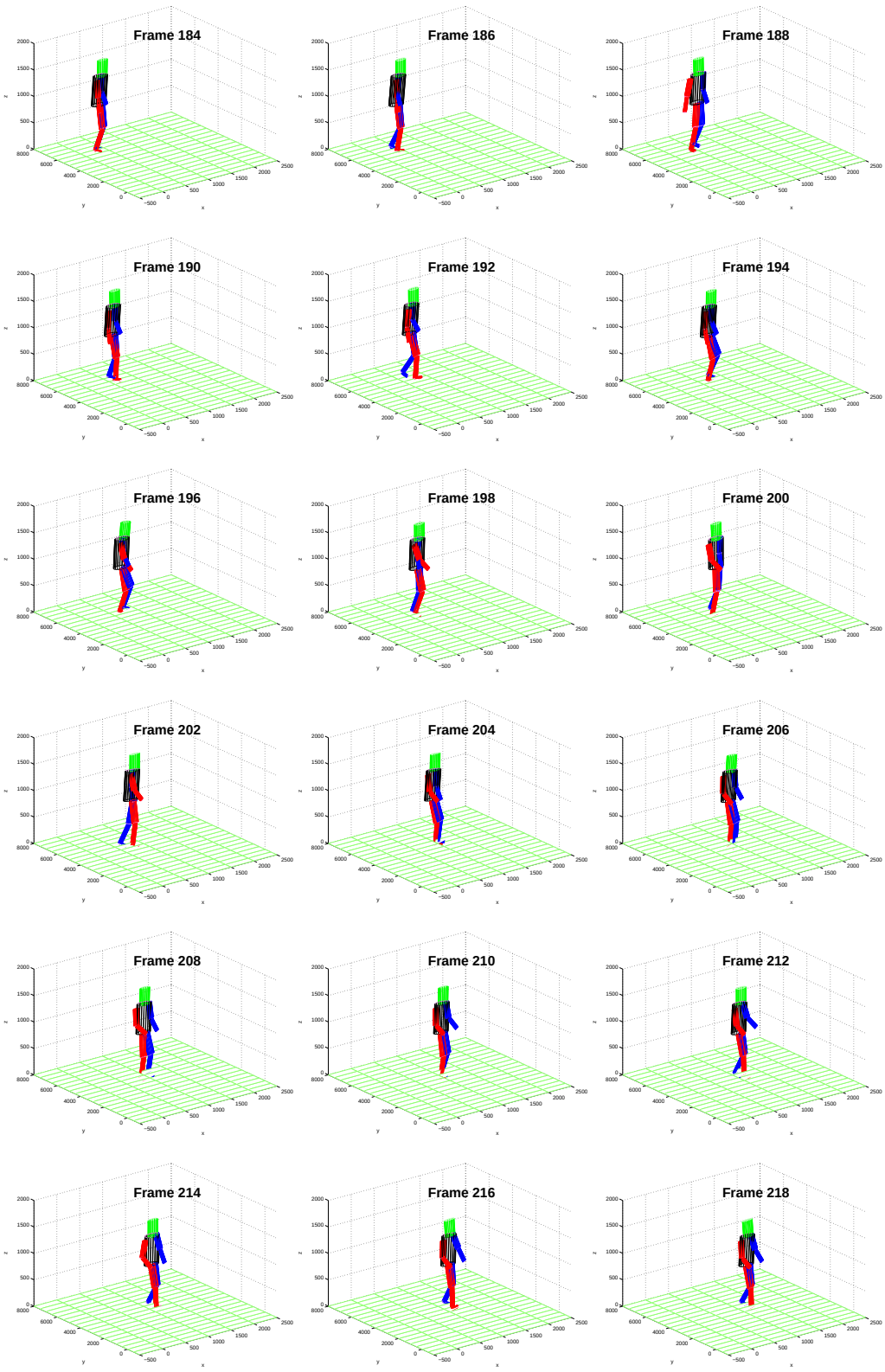



Figure 5.30: CAVIAR EnterExitCrossingPaths1 3D reconstruction with every second frames in the range of 184–218 [◊].

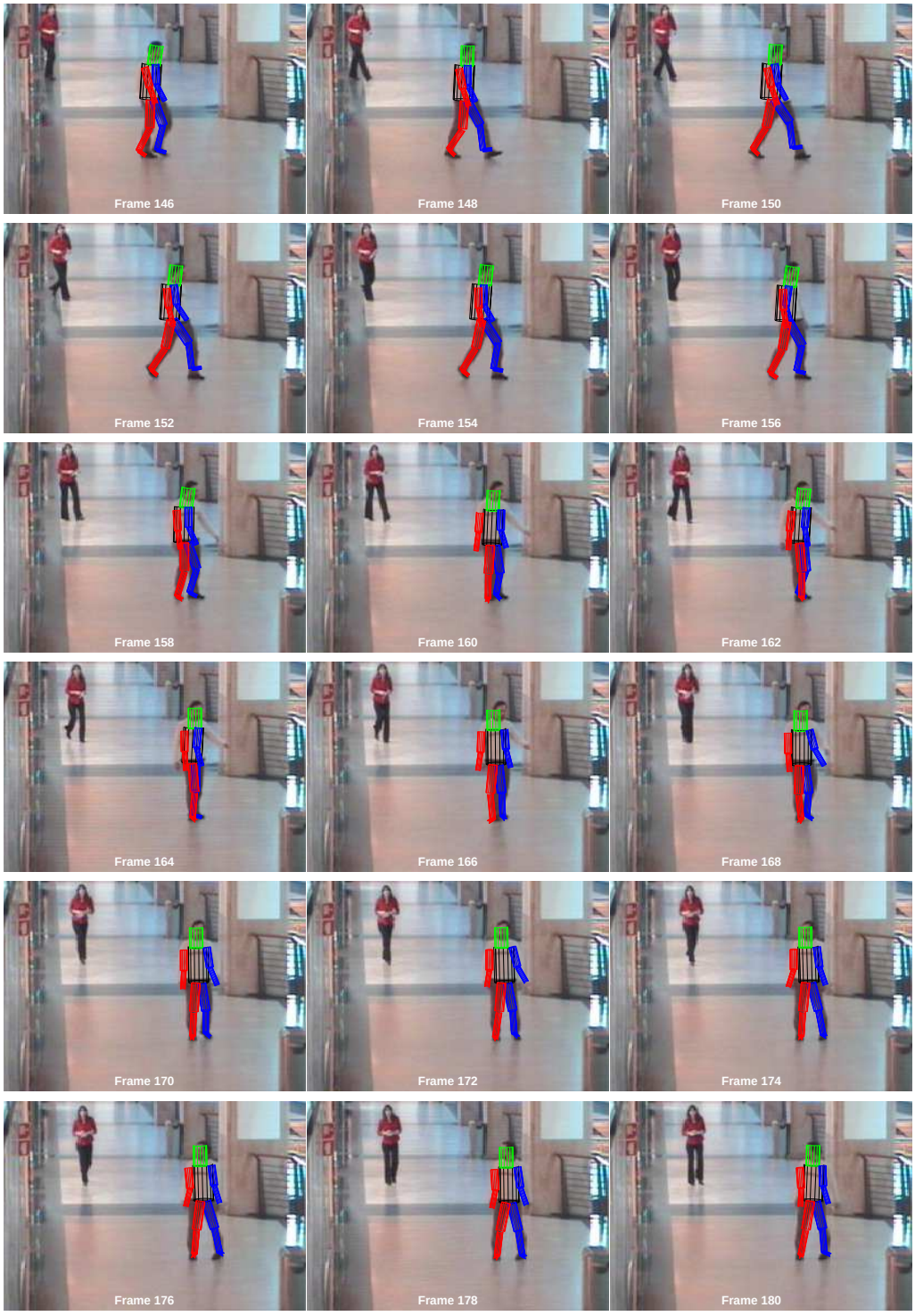



Figure 5.31: CAVIAR OneLeaveShopReenter1 corridor sequence I with every second frames in the range of 146–180 [◊].

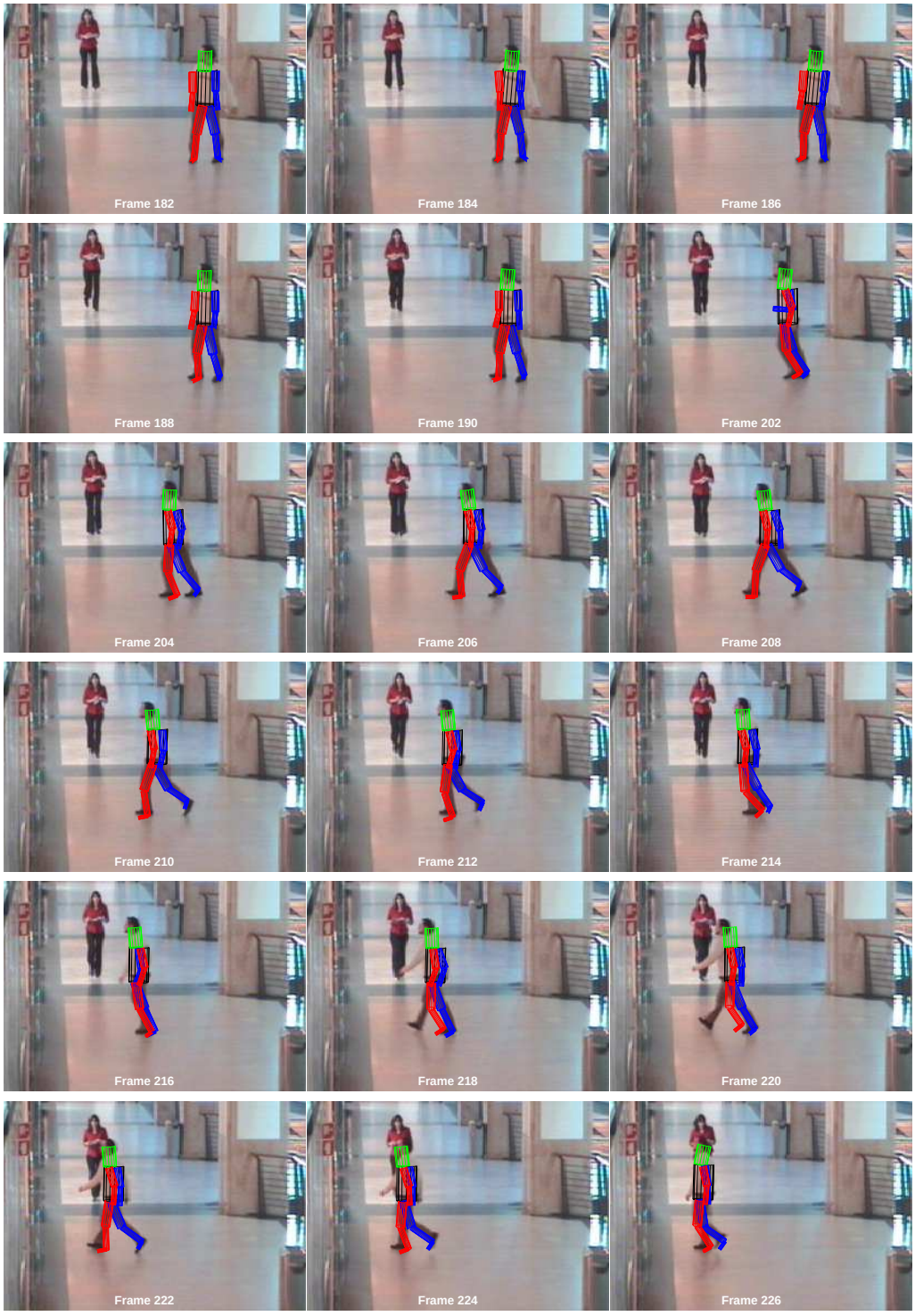



Figure 5.32: CAVIAR OneLeaveShopReenter1 corridor sequence II with every second frames in the range of 180–226 [◊].

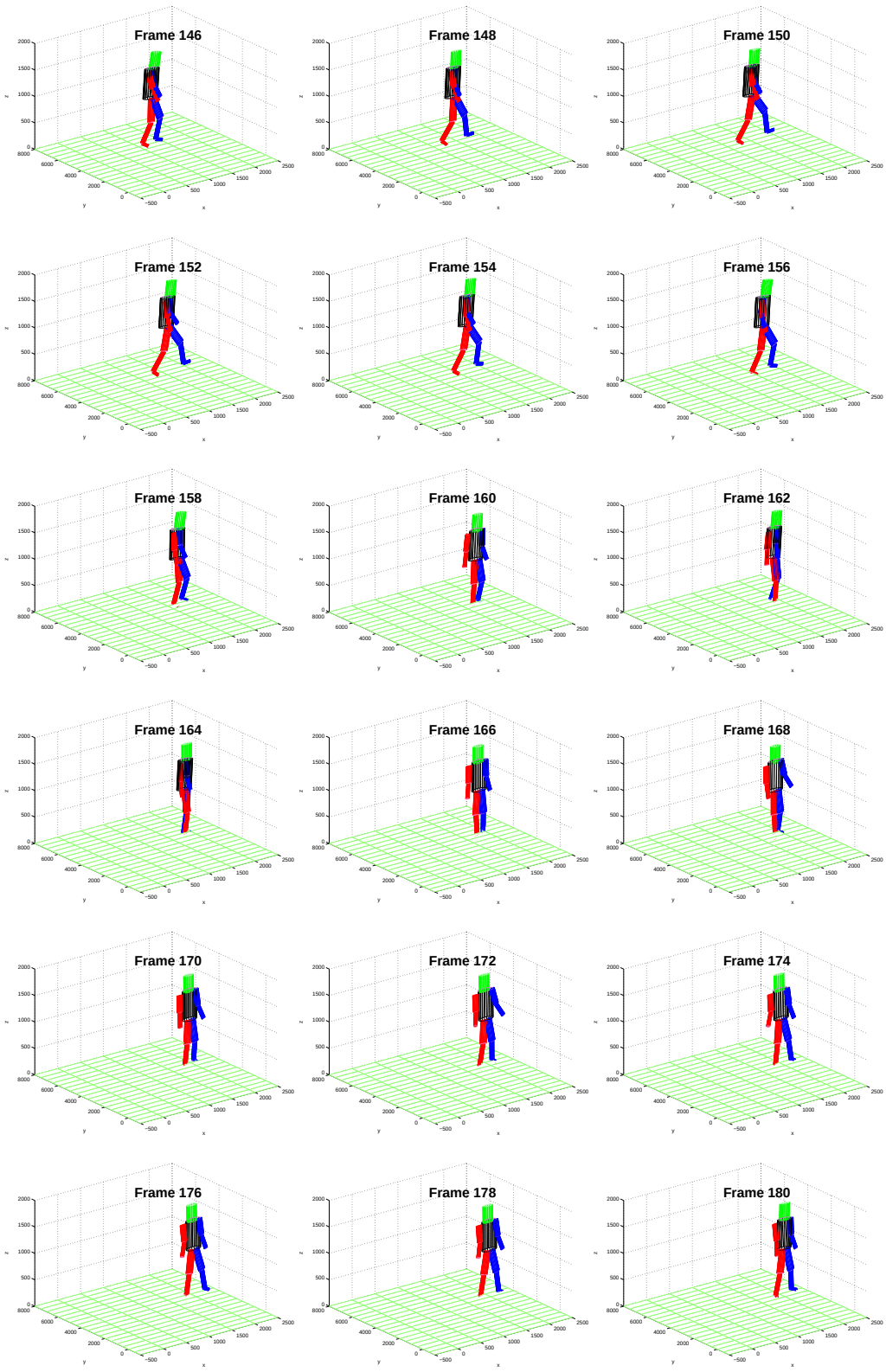



Figure 5.33: CAVIAR OneLeaveShopReenter1 3D reconstruction I with every second frames in the range of 146–180 [◊].

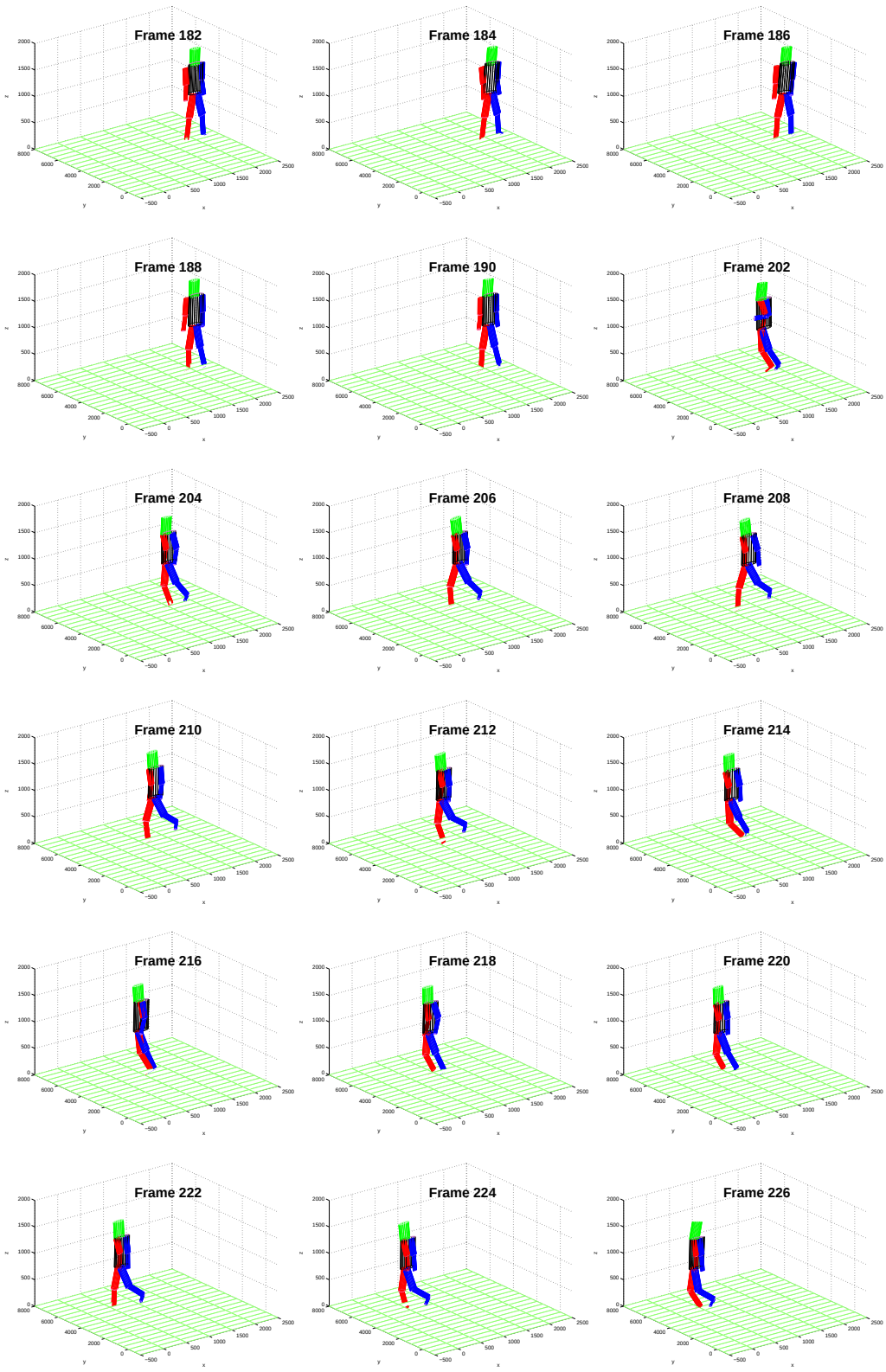



Figure 5.34: CAVIAR OneLeaveShopReenter1 3D reconstruction II with every second frames in the range of 180–226 [◊].


Figure 5.35: i-LIDS AVSS AB Easy sequence 2D view with frames 1075–1092

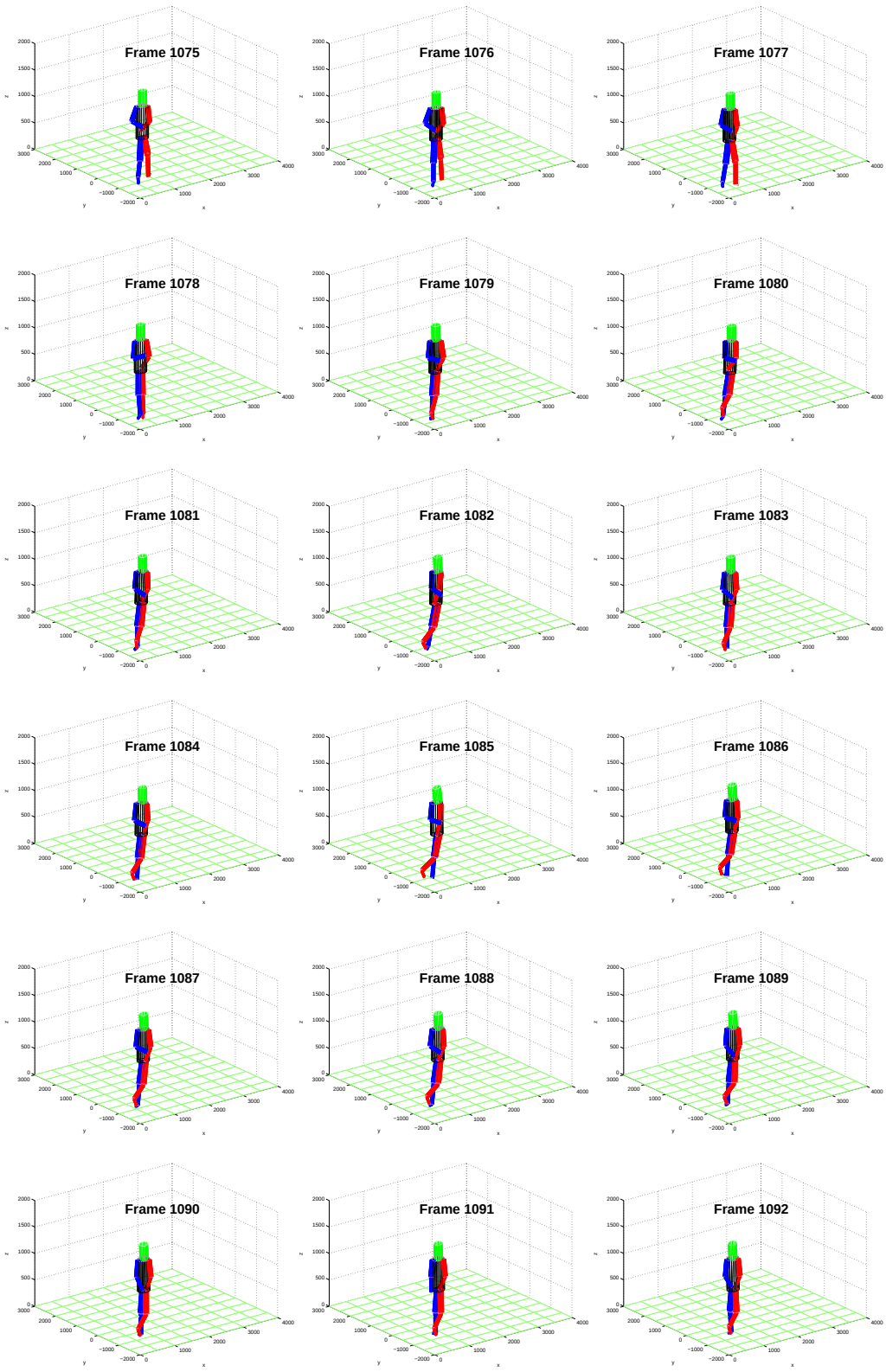



Figure 5.36: i-LIDS AVSS AB Easy sequence reconstruction with frames 1075–1092 [◊].

## 5.7 Summary and conclusions

The *Hierarchical Partitioned Particle Filter* tracks articulated, high dimensional structures with hierarchical dependence between some parameters and independence between others.

The HPPF was evaluated against the basic PF with SIR and the APF. Compared to these, it shows improved tracking for both visual and quantitative evaluation.

With HPPF-MCM, particles model movements and recover the current pose. As a motion model, the MCM provides switching motion modes on each level of the tracker. It was shown how this further reduces the tracking error.

The main disadvantages of HPPF-MCM, similar to other trackers, are the several tuning parameters. If they are optimised then tracking results are good, however this is laborious. The optimisation of one or a limited set of parameters at a time, and the effects on tracking performance, were presented. This concluded that the required particle number is low, not more than 200 particles; the estimate based on the proposed windowed-mean is advantageous; the combination of several likelihoods is beneficial, however the colour and edge components have lower importance than one would expect; and the weight postprocessing increases the tracking performance. Surprisingly, as chapter 4 also observed, MCMs with longer movements or with more MC do not improve the tracking, but the two have adverse effects. The above effects of the tracking parameters on the HPPF-MCM tracking are summarised in table 5.21.

The HPPF-MCM was evaluated on the HumanEva I and II datasets. The tracking on unseen (*e.g. S4 Combo 4*) sequences is accurate, therefore it is expected that the HPPF-MCM performs equally well on any other HumanEva sequence. It has to be emphasised that motion is not constrained to the activities present in the training set.

The analysis on HumanEva sequences reducing the input from three to two and one cameras shows performance degradation and failure for monocular sequences. Further, the two camera CAVIAR sequence is well tracked, while i-LIDS fails with a single camera.

The HPPF tracks the 20.9s long *S4 Combo 4* sequence, with transitions between *Walk*, *Jog* and *Balance* activities, without signs of performance degradation, while APF was proved only on short, 4–6s long, good contrast, video data.

The tracking and learnt motion model was primarily tested on the HumanEva dataset, however after certain scene specific initialisations (*i.e.* camera calibration, tracked person size and position definition) the HPPF-MCM tracker is expected to work on other se-

| **Parameter** | **Effect** |
|---|---|
| number of cluster ($n_{\mathcal{C}}$) and length of movement ($l_m$) | have adverse effects, optimum is at the middle of their range, with clusters not overspecialised, but unambiguous; movements have to be long enough to be specific, but not in excess, that out averages their characteristics; |
| propagation mode | a multi-modal propagation with multiple motion modes is better compared to single mode, however mixing them is heuristic; |
| propagation mode constants | one standard deviation is optimal for pose, random pose and speed modes, while normal model can be reduced to a jump to the mean pose; current constants may be sensitive on the used training data; |
| number of particles ($n_p$) | more particles are describing better the underlying distribution, however a limited number (*i.e.* 200) particles of the HPPF provide better results than PF or APF with equivalent number of particles; |
| tracking estimate | Windowed-mean offers better estimate in a multi-modal distribution, unfortunately for other applications the definition of the window might not be obvious; |
| particle survival | scaling and low weight elimination enhance the tracking; however their general applicability is not proved; |
| likelihood and priors | domain knowledge and additional measurements improve the tracking; construction of such needs expertise; |

Table 5.21: Summary of tracking effects of the HPPF-MCM parameters

quences. This is achieved by means of 3D modelling, multi-modal motion model and sampled, equalled frame rate, but also by the stochastic components of the motion model. However, occlusions by static or dynamic objects, and environmental changes with low quality observations may need modifications of the algorithm. Experiments on CAVIAR, i-LIDS and also on unseen HumanEva sequences demonstrated this generalisation. Tuning the motion model, *e.g.* for better i-LIDS, with constrained motion to the dominant *Walk* may also result in a performance gain.

Since the main effort of HPPF design required the parameter optimisation, this is the major flaw of the filter, however other parametric filters suffer from the same flaw.

# Chapter 6

# Combining tracking and behavioural analysis

For behavioural understanding with model recovery, the action analysis and the articulated human tracking were defined and analysed independently in chapters 4 and 5. Next, these results are combined together for a whole tracking-behavioural system.

An objective evaluation of complex activities implies several exhaustive tests on a large ground truth dataset, labelled by several independent human observers. Because the costs of acquisition, labelling and distribution are prohibitive, such a dataset is not available. Therefore the evaluation results of this chapter are limited to subjective analysis of sequences from the HumanEva and CAVIAR datasets.

The chapter first connects the behavioural analysis with the *Hierarchical Partitioned Particle Filter* (HPPF). The tracking (for prediction) and the behaviour analysis use identical or different movement models. The effect of both on recognition is evaluated, followed by the analysis of the model parameters.

It also examines recognition with different partitions of the pose parameter. Finally, the recognition results of several sequences are presented and discussed.

## 6.1 Behaviour from the tracked model

First in order of processing, articulated human tracking (chapter 5) recovers the probability distribution of the model from the input images or a video sequence and represents it by

the set of the particles, $\Psi_t$:

$$\Psi_t = \{\mathrm{p}_t(i)\}_{i\in 1\ldots n_p} = \texttt{Tracking}(\{\mathrm{O}_\tau\}_{\tau\leq t}). \tag{6.1}$$

Then, the behavioural analysis (chapter 4) recovers action labels using a *Movement Cluster Model* (MCM), $\mathcal{M}$. This model can be integrated into higher level interpretation. The model movements are defined by any *Body Feature Vector* (BFV) with the $\phi$ partition of the *Pose Vector* (PV). An action label $l$ for the movement m from equation (4.21) has probability:

$$\mathcal{L}_l(\mathrm{m}) = \texttt{Analysis}(\mathrm{m}) \tag{6.2}$$

$$= \sum_{\mathcal{C}} \mathcal{P}(l|\mathcal{C})\mathcal{P}(\mathcal{C}|\mathrm{m}), \tag{6.3}$$

The relationship of `Tracking` in video images for the intermediate human model, and of `Analysis` of this for symbolic description, is depicted in figure 6.1.

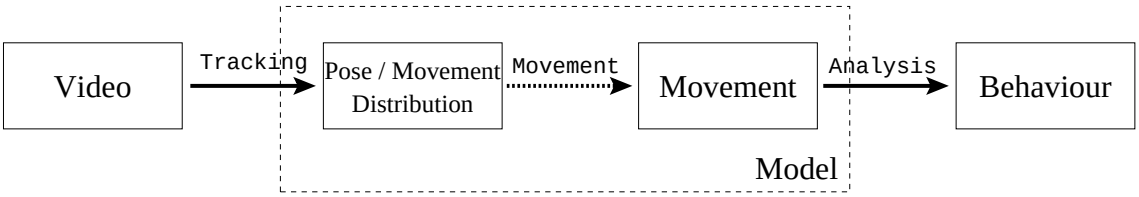



Figure 6.1: Tracking and behavioural subsystem integration. The video data is transformed by `Tracking` into pose or movement distribution that provides movements for `Analysis` for action labels.
.

`Tracking` provides distribution of the recovered model (*i.e.* pose or movements), while `Analysis` requires movements as input. For completeness, the function

$$\mathrm{m}_t = \texttt{Movement}(\{\Psi_\tau\}_{\tau\leq t}, \phi) \tag{6.4}$$

links the two and extracts the movement $\mathrm{m}_t$. For this, two alternatives follow next.

First alternative, since the particles in the HPPF-MCM tracker are movements (*i.e.* PVs with a history), the movement distribution is the current particle distribution. Therefore,

$$\texttt{Movement}_1(\{\Psi_\tau\}_{\tau\leq t}, \phi) = \Psi_t^\phi, \tag{6.5}$$

where

$$\Psi_t^{\phi} = \{\mathrm{p}_t^{\phi}(i)\}_{i\in 1\ldots n_p}, \tag{6.6}$$

is the distribution with the movement parameters $\phi$ extracted from the particles. $\texttt{Movement}_1$ is a distribution of movements given directly by the particle distribution $\Psi_t$, with each particle having its history, *i.e.* is a movement. Therefore, the label probability from equation (6.3) can be computed as the expectation over the movement distribution that is equal with the particle distribution. Hence,

$$\mathcal{L}_l(\mathrm{m}) = \underset{i=1\ldots n_p}{\mathcal{E}} < \sum_{\mathcal{C}} \mathcal{P}(l|\mathcal{C})\mathcal{P}(\mathcal{C}|\mathrm{p}_t^{\phi}(i)) >, \tag{6.7}$$

is the probability of the label $l$.

With the second alternative, the movement is composed by conjoining the $l_m$ consecutive current BFVs of the $\overline{P}_t$ estimated particle (equation (5.38)) of the $\Psi_\tau$. The current BFV for the partition $\phi$ from the estimated particle is ${}_0\overline{\mathrm{p}}_{t-l_m+1}^{\phi}$, hence the current movement results in:

$$\texttt{Movement}_2(\{\Psi_\tau\}_{\tau\le t}, \phi) = \left[{}_0\overline{\mathrm{p}}_{t-l_m+1}^{\phi}, {}_0\overline{\mathrm{p}}_{t-l_m+2}^{\phi}, \ldots, {}_0\overline{\mathrm{p}}_{t}^{\phi}\right]. \tag{6.8}$$

This, with equations (6.1) and (6.3) defines completely the label probability $\mathcal{L}_l$.

## 6.2 The influence of the MCM parameters

The MCM successfully provides motion prediction and movement analysis. However, both are affected individually by the length $l_m$ of the movement and the number of movement clusters $n_{\mathcal{C}}$, as shown in sections 4.6.6 respectively 5.4.1.

For this section, the movement distribution propagates through the behavioural system, and equation (6.7) defines the probability of each action label.

The effects of $n_{\mathcal{C}}$ and $l_m$ on the joint tracking-analysis system are evaluated concurrently for $5\times 6$, $n_{\mathcal{C}}$ and $l_m$ values, on the *Walk*, *Throw/Catch*, *Gesture* and *Jog* sequences. Their comparison is shown by the confusion matrices, figure 6.2, and by the less subjective accuracies (defined by equation (2.18)), table 6.1.

Since resource limits restricted tracking tests to four out of the five HumanEva sequences, the confusion matrices are $4 \times 5$ with no *Box* activity. Also, recalling from

| $l_m$ | $n_{\mathcal{C}}$ | | | | |
|---|---|---|---|---|---|
| | 20 | 40 | 60 | 80 | 100 |
| 1 | 0.32 | 0.29 | 0.28 | 0.26 | 0.34 |
| 3 | 0.29 | 0.32 | 0.25 | 0.37 | 0.25 |
| 5 | 0.24 | 0.30 | 0.26 | 0.32 | 0.27 |
| 15 | 0.27 | 0.30 | 0.17 | 0.07 | 0.08 |
| 25 | 0.27 | 0.02 | 0.01 | 0.00 | 0.00 |
| 35 | 0.17 | 0.01 | 0.00 | 0.00 | 0.00 |

Table 6.1: Recognition accuracy with identical MCMs for both tracking and behavioural analysis. $\mathcal{M}_1$ MCM is used only for global action recognition.

section 4.6.3, matrices do not normalise to one, since labels are considered independent, each with probability between zero and one. Null-lines are possible, if none of the action cluster is recognised (*i.e.* this is the non-recognised action).

Random guessing of one out of the four plus an *unknown* action (that includes *Box*) results in an accuracy of 0.20. Accuracies, table 6.1, are above the this for shorter movements. However, for greater $l_m$, none of the independent actions is recognised and the accuracy is zero.

The accuracy decreases with $l_m$ and $n_{\mathcal{C}}$. This was already motivated by the increased distances in the high dimensional space from the *Movement Cluster* (MC) centres, caused by accumulated error with longer movements, and by the extra, more specialised, lower covariance clusters that result in distant movements from the clusters.

Compared to the test on perfect HumanEva data (section 4.6.3) recognition is weaker with the tracked data. This matches observations of section 4.6.6, which suggests a performance degradation with added error. However, the expected accuracy from table 4.6 for $\sigma = 0.8$ corresponding to 100 mm absolute error, is better (*i.e.* with values up to 0.8) than obtained from the tracked parameters. Differences are motivated by the uneven distribution of the errors on different HumanEva sequences (see table 5.17), the different errors in parameters (*i.e.* lower compared to upper-limb parameters are error-prone), both assumed equal (*i.e.* $\sigma$) and by the missing *Box* activity.

The best recognised action is *Throw/Catch* (figure 6.2). The other actions are confused with *Throw/Catch* for shorter movements ($D \leq 5$) and with *Gesture* for longer movements ($D \geq 15$).

Since tracking is not accurate, recognition shifts towards the most generic action as $l_m$ increases; for the dataset trained, this is *Gesture*.

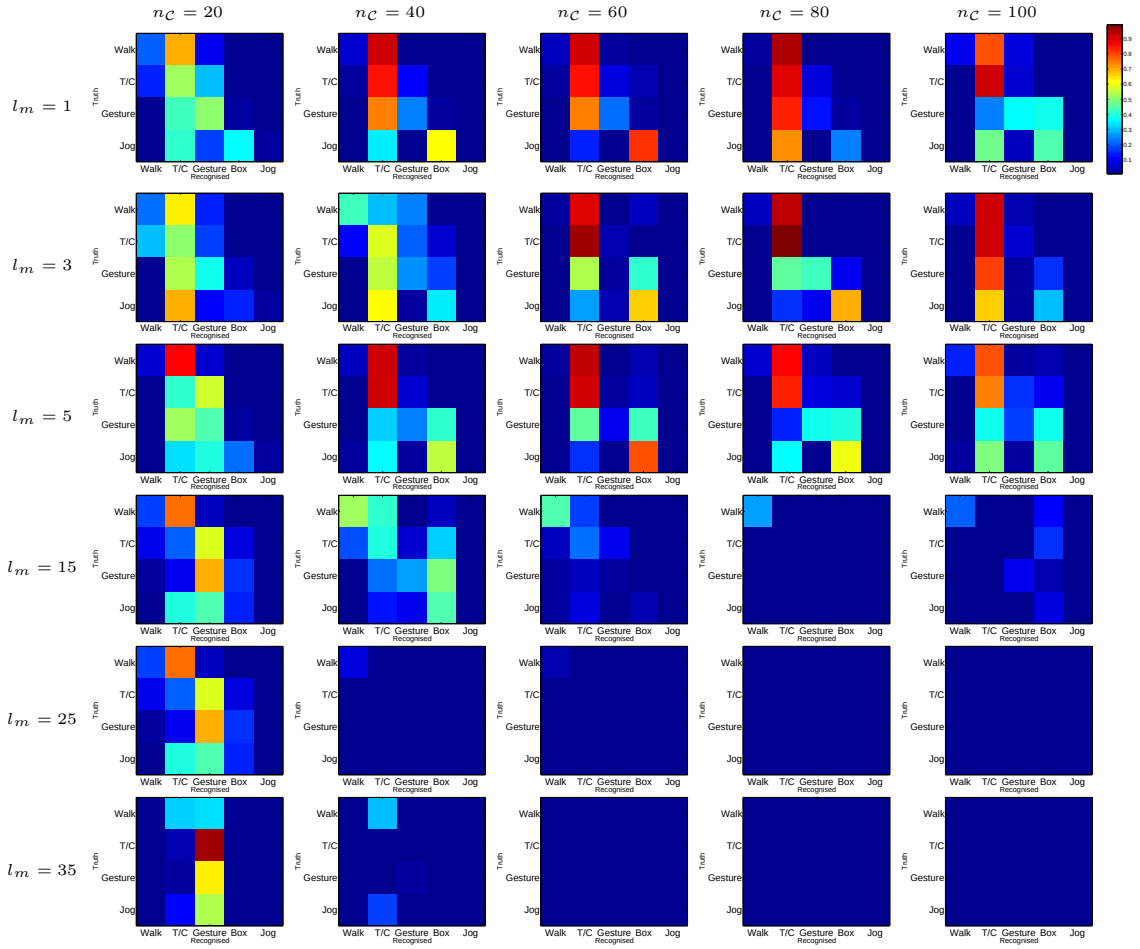



Figure 6.2: Confusion matrix dependence on number of clusters $n_{\mathcal{C}} = 20, 40, 60, 80, 100$ and length of sequence $l_m = 1, 3, 5, 15, 25, 35$ of the MCM parameters for $\texttt{Movement}_1$, *i.e.* the same MCM used both for visual tracking and behavioural analysis of the tracked data.

The best recognitions are obtained for $n_{\mathcal{C}} = 80$ and $l_m = 3$ or $l_m = 5$. Since $n_{\mathcal{C}} = 80$ and $l_m = 5$ also provides the lowest tracking error (table 5.8) this suggests that good tracking supports action recognition.

## 6.3 Tracking and analysis with independent models

With the formulation of equation (6.8), the MCM parameters for behavioural analysis are independent of the MCM used in HPPF for motion prediction. Therefore MCM analysis dependence on $n_{\mathcal{C}}$ and $l_m$ is further analysed using the same, HPPF-MCM tracker with lowest errors ($l_m = 5$, $n_{\mathcal{C}} = 80$). The accuracies, shown in table 6.2, are marginally poorer than for the HPPF integrated model, however they show the same degradation with both $l_m$ and $n_{\mathcal{C}}$. As in table 6.1, the accuracies are highest with $n_{\mathcal{C}} = 80$ and $l_m = 3$ or $l_m = 5$.

| $l_m$ | $n_{\mathcal{C}}$ | | | | |
|---|---|---|---|---|---|
| | 20 | 40 | 60 | 80 | 100 |
| 1 | 0.30 | 0.34 | 0.31 | 0.34 | 0.35 |
| 3 | 0.31 | 0.23 | 0.28 | 0.35 | 0.26 |
| 5 | 0.27 | 0.29 | 0.22 | 0.31 | 0.33 |
| 15 | 0.29 | 0.32 | 0.25 | 0.08 | 0.04 |
| 25 | 0.23 | 0.03 | 0.00 | 0.00 | 0.00 |
| 35 | 0.18 | 0.01 | 0.00 | 0.00 | 0.00 |

Table 6.2: Recognition accuracy for independent MCMs for both tracking and behavioural analysis. $\mathcal{M}_1$ MCM is used only for global action recognition.

Comparing tables 6.1 and 6.2, one concludes that only minor differences exist in accuracies for identical respectively independent MCMs used for tracking and for analysis. For both the accuracies are low. Increasing either movement length $l_m$, or cluster number $n_{\mathcal{C}}$, degrades recognition. If both parameters are increased concurrently, there are improvements, limited however to $n_{\mathcal{C}} \leq 5$. Next, only the formulation of equation (6.7) is used to compute the label probability.

## 6.4 The influence of the MCM detail

The above results for activity recognition with the full pose MCM, $\mathcal{M}_1$ are poor. However the models $\mathcal{M}_i$ from table 4.3, with reduced BFVs, tackle different subsets of the parameter space and provide different recognition results. The recognition using the MCM $\mathcal{M}_7$ (*i.e.* left upper arm parameters) has the accuracies shown in table 6.3.

| $l_m$ | $n_C$ | | | | |
|---|---|---|---|---|---|
| | 20 | 40 | 60 | 80 | 100 |
| 1 | 0.26 | 0.26 | 0.27 | 0.26 | 0.26 |
| 3 | 0.26 | 0.25 | 0.27 | 0.28 | 0.28 |
| 5 | 0.30 | 0.29 | 0.29 | 0.31 | 0.31 |
| 15 | 0.31 | 0.41 | 0.39 | 0.38 | 0.34 |
| 25 | 0.31 | 0.34 | 0.40 | 0.36 | 0.35 |
| 35 | 0.35 | 0.38 | 0.33 | 0.46 | 0.42 |

Table 6.3: Recognition accuracy with $\mathcal{M}_7$ MCM is used only for global action recognition.

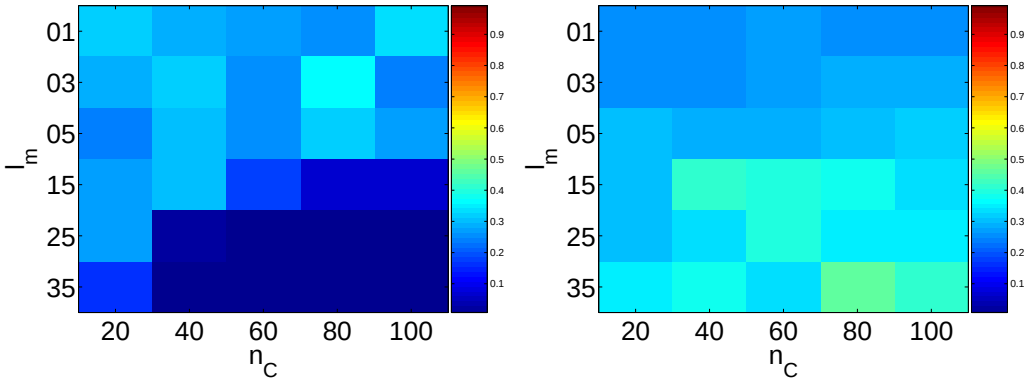



Figure 6.3: Accuracies for $\mathcal{M}_1$, full body and $\mathcal{M}_7$, left upper arm partitions. Figures are identical with tables 6.2 and 6.3.

Side by side, the recognition accuracies of $\mathcal{M}_1$ and $\mathcal{M}_7$ (figure 6.3) suggest that a partition of parameters recognises actions better then the whole set; for models with smaller partitions (*i.e.* $\mathcal{M}_7$), a higher value of $l_m$ results in better recognition. The first observation is motivated by the lower dimensionality parameter space of the limb compared to the full pose MCM (*i.e.* two against 18 dimensions). Since the longer MCs capture longer motion dynamics, better recognitions of the longer actions with longer movements explain the second observation. It is also observed that the effect of increased MC number is not visible using either of the models.

## 6.5 Recognition of HumanEva sequences

As well as global, quantitative evaluation of accuracy, detailed frame-by-frame analysis provides an insight into the recognition result, although it is subjective and qualitative. Figures 6.4, 6.6 and 6.7, like the diagrams in chapter 4, visualise for the *S1 Walking 1*, *S1 Gesture 1* and *S1 Jog 1* sequences the probability of the labels (horizontal axis) for each frame (vertical axis). The 13 diagrams for each sequence correspond to recognition

with one of the $\mathcal{M}_i$ MCMs (the recognition with the head MCM, being ambiguous, was again omitted). Both the tracking and the behavioural analysis use the same MCM, with $l_m = 5$ and $n_{\mathcal{C}} = 80$.

The *S1 Walking 1* with whole body MCM recognises correctly the *Walk* action in the initial input, until frame 18, and in frames 45–72, while the other 90% of the frames have higher *Throw/Catch* probabilities. However, the diagrams show that six out of twelve local MCMs (whole lower left and right arms, right upper arm, left upper left and right lower arm) produce high walking probabilities. The other MCM fail, and recognise *Throw/Catch* or *Box*.

In addition to the global action labels, the local labels provide detailed description of the action. These are evaluated either visually, comparing them frame-by-frame to the image, or qualitatively, by their periodic alternation. The repetitive patterns of *Right stride back* and *front*, *Left stride back* (least visible) and *front* are the best visible on the *Left upper leg* MCM. The anti-phase relationship of *left arm forward*, *right arm backwards* and *right arm forward*, *left arm backwards* is also visible in this diagram.

Figure 6.5 shows the labels recovered from the start of the *S1 Walking 1* sequence using the $\mathcal{M}_1$ MCM superimposed with the input *S1 Walking 1* sequence frames. For clarity, labels are grouped into *General*, *Arm (left/right)* and *Leg (left/right)* semantics. Only labels with probability above 0.5 are displayed, in blue, and those with above 0.8, in green.

The *Walk* action is recognised in 11 frames, while it is misclassified as *Throw/Catch* in 7 frames. Low-level labels are correctly detected without misclassifications, however in frames classified as *Throw/Catch*, the low level labels are not detected. This was expected, since labels are attached to MCs; *Throw/Catch* sequences, and therefore *Throw/Catch* MCs, were not trained with arm and leg actions.

For the *S1 Gesture 1* sequence, figure 6.6, the gesture actions are recognised with high probability on the majority of the levels. With $\mathcal{M}_1$, movements are labelled well for almost half of the sequence, however the later movements are considered *Box*, being both static and arm related actions.

The confusion matrices (figure 6.2) already signalled that the *Jog* sequence classification fails in the majority of the cases. These are the components with the least accuracy. The diagrams of the *S1 Jog 1*, figure 6.7, confirm this, since none of the levels have relevant *Jog* recognition probability. The recognised actions are *Walk*, *Throw/Catch*, *Box*

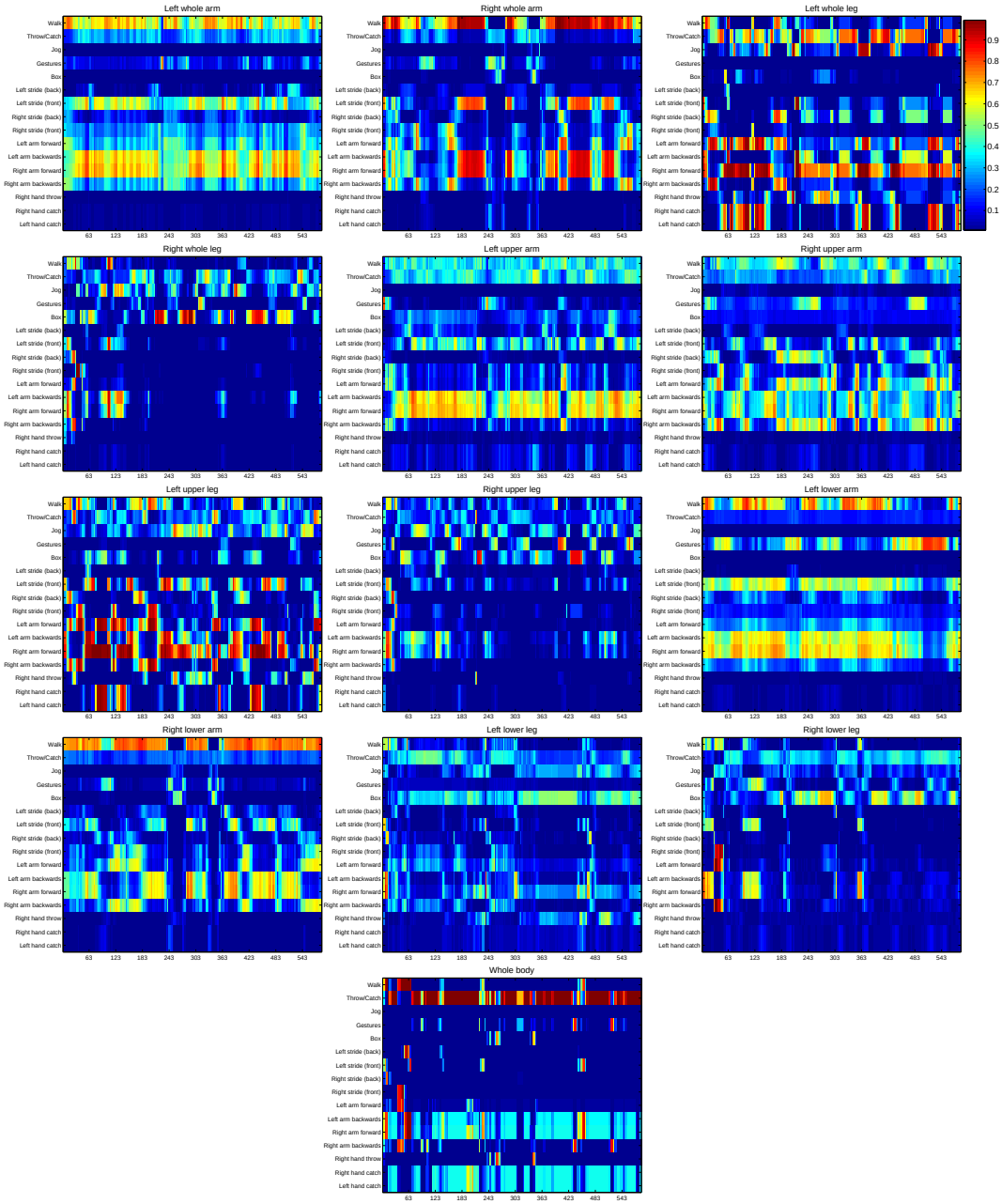



Figure 6.4: HumanEva *S1 Walking 1* sequence recognition. On each of the 13 MCM levels, for all frames (on horizontal) the probability of each label (vertical) is shown colour coded.

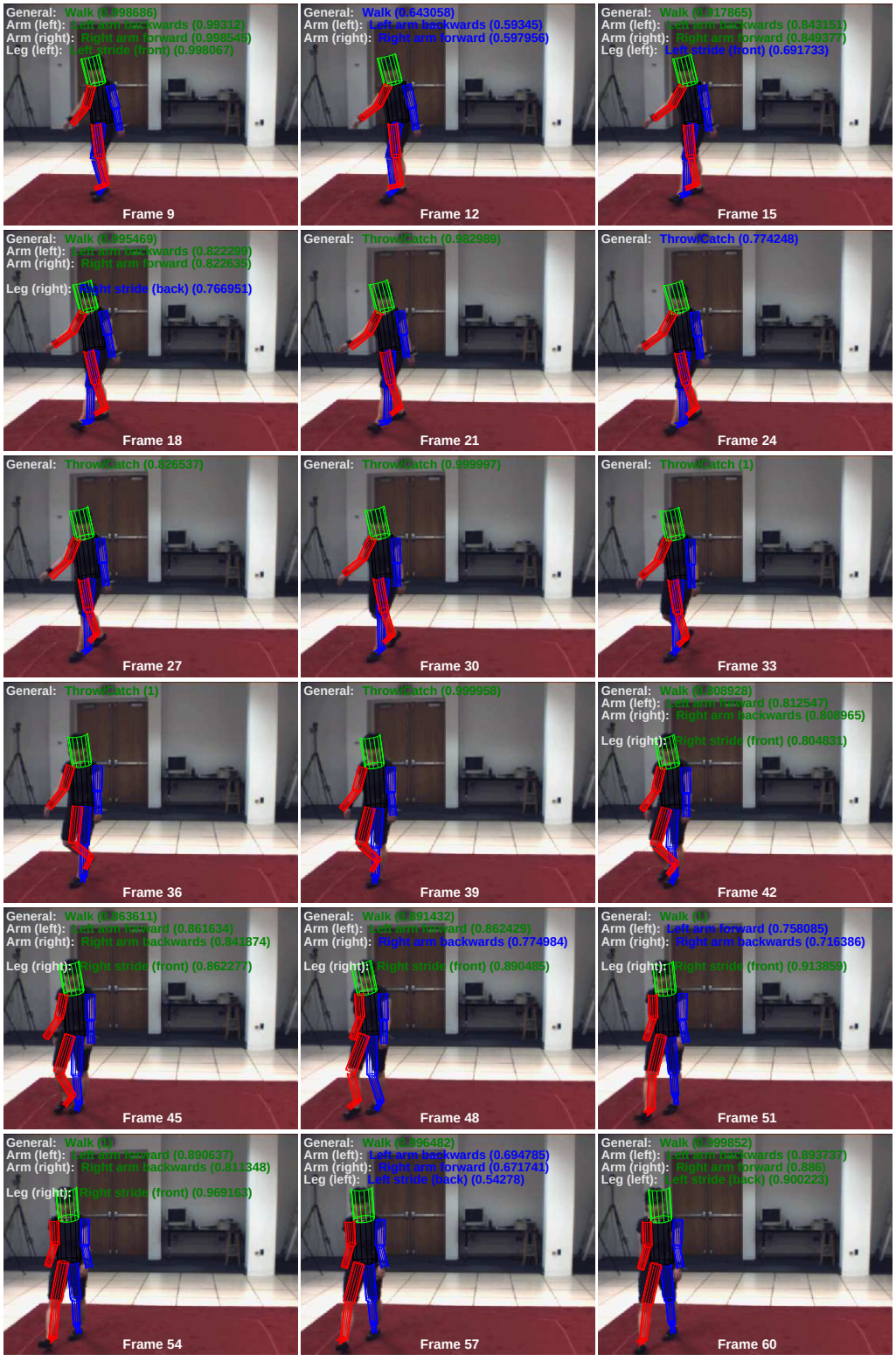



Figure 6.5: The recovered HumanEva *S1 Walking 1* labels superimposed with the input frames. Labels are manually grouped on different lines into *General*, *Arm (left/right)* and *Leg (left/right)* semantics. Only labels with probability above 0.5 are displayed, in blue, and those with above 0.8, in green [◊].

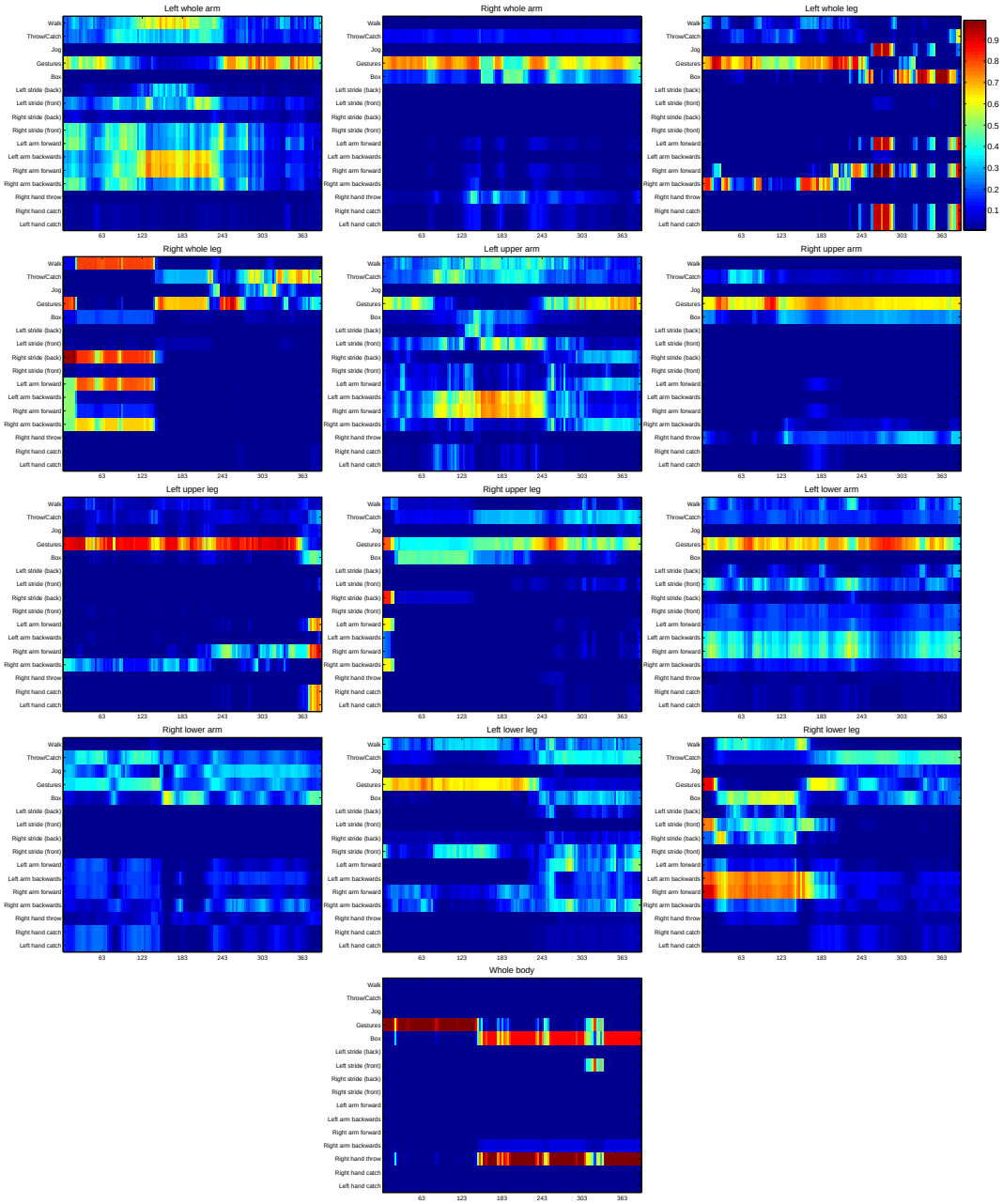



Figure 6.6: HumanEva *S1 Gesture 1* sequence recognition. On each of the 13 MCM levels, for all frames (on horizontal) the probability of each label (vertical) is shown colour coded.

and *Gesture*. This can be explained by the similarity of *Jog* to all of these actions, while distinctive features of the *Jog* (*e.g.* the body translation speed) are ignored.

## 6.6 Recognition with reduced camera number

Behavioural recognition depends on the accuracy of the tracking, which in turn depends on the number of cameras. The four scenarios are: all three available cameras C1, C2 and C3; only two cameras, C1 and C2; C1 alone; C2 alone. These result in accuracies of 32%, 23%, 22%, and 30% respectively (figure 6.8). This verifies that fewer cameras are less accurate. However, surprisingly, the difference between the cases with all three cameras and with C2 only is small. This might suggest, that profie views (*i.e.* C2 has longer profile views than C1) are more important for the behavioural analysis than front views.

## 6.7 Recognition of the CAVIAR sequence

The CAVIAR tracking from chapter 5 is not stable, and several frames have visual tracking errors. This compromises recognition of both whole body movements and longer movements. Figure 6.9 shows the classification for each MCM. Similar to the HumanEva *Walk* sequence (figure 6.5), the full pose MCM is not effective to recover the *Walk* actions. As for the *S1 Walking 1*, the *Left upper leg*, $\mathcal{M}_9$, MCM provides the most detailed information about the visible periodic motion patterns, while seven levels recognise *Walk* with higher probability than other actions.

With this MCM, figure 6.10 visualises the action labels, superimposed with the first 18 frames of the tracked sequence. Again, there are frames misclassified with the similar but static *Throw/Catch* activity. However, the rest of the frames are classified as *Walk*, and include local action descriptions.

## 6.8 Discussion

This chapter connects the tracking and behavioural analyses, and tests the final output of the joint system. The separation, argued before, allows modularity and flexibility, and incremental abstraction from the input data, while maintaining the probabilistic modelling until the output behaviour.

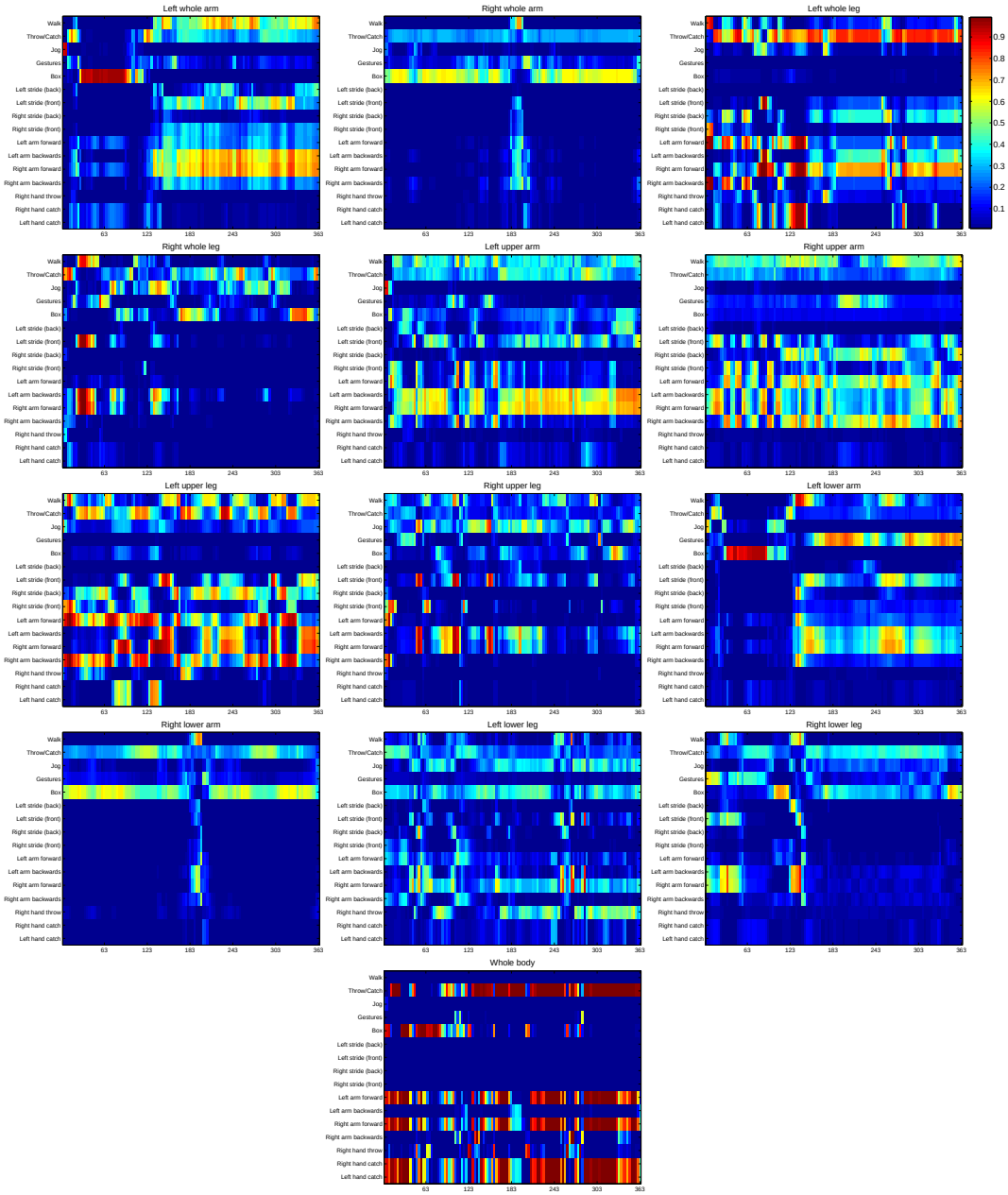



Figure 6.7: HumanEva *S1 Jog 1* sequence recognition. On each of the 13 MCM levels, for all frames (on horizontal) the probability of each label (vertical) is shown colour coded.

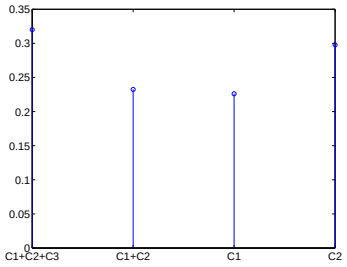



Figure 6.8: Recognition rate with reduced camera number.

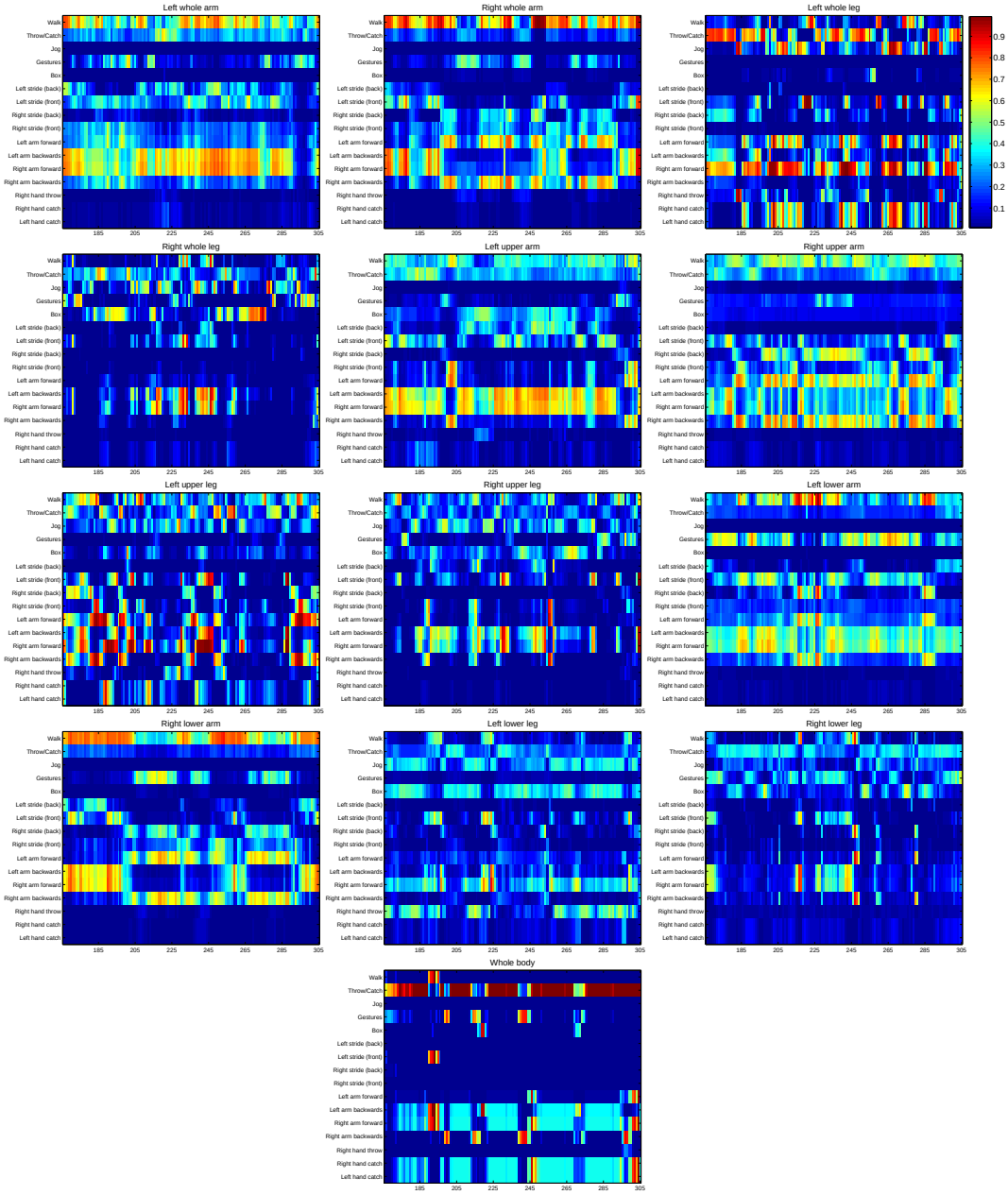



Figure 6.9: CAVIAR *EnterExitCrossingPaths1*. On each of the 13 MCM levels, for all frames (on horizontal) the probability of each label (vertical) is shown colour coded.

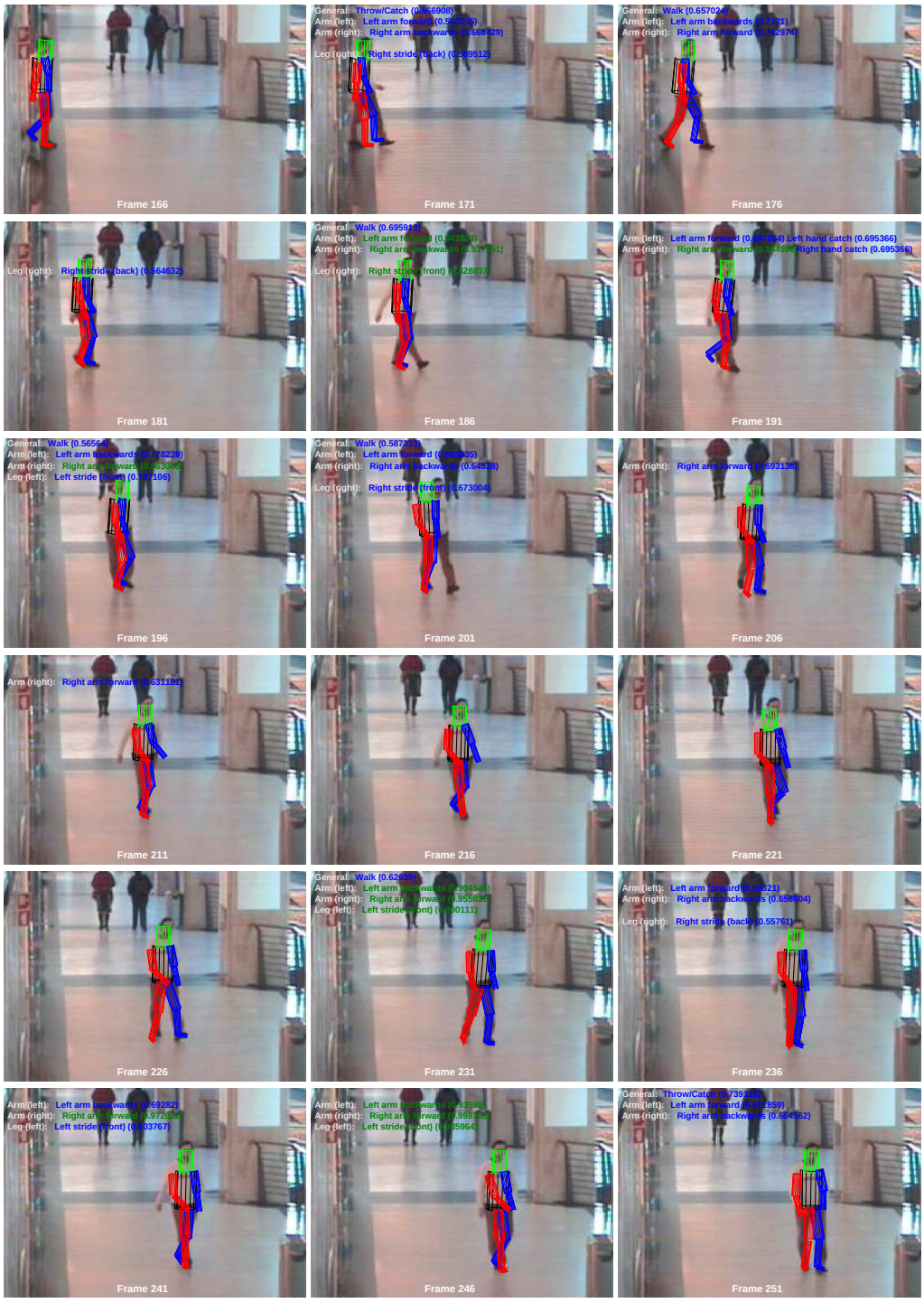



Figure 6.10: The recovered CAVIAR *EnterExitCrossingPaths1* labels superimposed with the input frames. Labels are manually grouped on different lines into *General*, *Arm (left/right)* and *Leg (left/right)* semantics. Only labels with probability above 0.5 are displayed, in blue, and those with above 0.8, in green [◊].

The movements are recovered by the tracker, then they become symbolically labelled actions. Examples with accurate action classifications were given. However, suggested by the confusion matrices, the approach is weak in classifying activities (*i.e.* whole sequences) by the majority vote of individual actions. This is because activities have a multitude of action components, and defining actions of the activity might not be the most frequent ones (*e.g.* *Throw/Catch* is best defined by the short throwing and catching actions and not the most frequent standing). Moreover, similar movements, *e.g.* all those that are classified as standing action, are part of different actions. Also, due to model recovery errors, severe problems are misclassifications of actions, or failure to classify them at all.

Further, while parameters describing the whole pose are generally not effective for discrimination, different partitions of the parameter space with the corresponding MCMs do perform better.

It was shown that the MCM based analysis provides detailed action symbols of the activity. The tests on HumanEva and the CAVIAR sequences prove that this detailed description is recovered.

The behavioural analysis requires good articulated tracking, and if single camera views are poor for this, these sequences (*e.g.* i-LIDS) can not be analysed.

Further, it must be mentioned that only articulated pose parameters were used for the analysis. Positional or velocity parameters are highly discriminative features of *Walk* and *Jog* and distinguish between static and moving activities, however they were not used in the described tests. The HPPF recovers well the position, however the behavioural analysis was built on a dynamical model for pose prediction. Since pose does not include positional parameters, the global body position was omitted also from behavioural analysis. Similarly, speed of parameter changes was also omitted, for the consideration that the change is explicitly encoded by the multiple poses forming a movement. However, parameters of BFV can arbitrary include positional or velocity parameters. While behaviour analysis would benefit from an extended BFV, the particles with larger dimensionality would increase tracking complexity and therefore this was avoided. Independent models trained for tracking and analysis may overcome this antagonistic setup.

Overall results from this chapter are limited, however they prove that the tracking and behavioural analysis can be separated, the MCM provides motion tracking for the HPPF, while it also performs behavioural analyses. It is able to recognise both global and low level actions. However, details of how to obtain the optimal model should be the subject

of further research.

# Chapter 7

# Conclusions and future work

This final chapter summarises the contributions of the thesis, the flaws of the methods presented, and the possible future research directions.

## 7.1 Summary and contributions

This work has focused both on articulated human tracking and on behavioural analysis of the recovered human motions. The contributions are in prior information modelling, articulated tracking and behavioural analysis.

- **Articulated human modelling**

  Chapter 3 defined the articulated human model, resembling a simplified human anatomy. Similar 3D models exist, but the chapter is novel in the definition of the likelihood functions that evaluate this model. Unlike other work, these can account for multiple camera views, individual body parts and multiple types of measurements. The chapter also presents an effective methodology for manual post-calibration of raw images.

- **Dynamic models**

  Chapter 4 defined three dynamic models for articulated objects. The *Pose Transition Model* (PTM) is similar to HMMs, but the representation of poses is compressed, and similar poses generate clusters.

  The *Continuous Time Pose Transition Model* (CTPTM) was introduced. This extends the initial PTM with explicit representation of the transition time. Therefore

the CTPTM models smooth transitions between discrete poses.

Further, the *Movement Cluster Model* (MCM) introduces movements as states of the model, and so overcomes the limitations of discrete pose representation, and defines transitions between states by a Gaussian-modelling of the resulting pose of each movement.

All three models were learnt unsupervised, and generated synthetic human motion.

- **Articulated human tracking**

  The *Hierarchical Partitioned Particle Filter* (HPPF), from chapter 5, was designed to track high dimensional structures with hierarchical dependence between some parameters, and independence between others. It is a generalisation of the partitioned and the annealed particle filter. Compared in section 5.3.4 to the basic or the annealed particle filter, it tracks better the human position and pose. It has been used with the dynamic model of the MCM, with motion prediction on full pose and multiple limb levels. The partition dependent likelihoods reflect only specific parameters, and therefore contribute to better overall performance of the HPPF.

  The HPPF includes enhancements of the particle distribution and estimate computations applicable in other PFs. First, the estimate of the particle distribution is not the mean or the mode, but their combination, the windowed-mean. Weighting normalisation and elimination also improve the tracking.

  The flaw of the tracker is that several parameters require optimisation. However, this has been bypassed by systematic optimisation.

- **Behavioural analysis**

  Chapter 4 introduced behavioural models with MCMs. Activity labels were attached to movements, with supervised training that followed the *Movement Cluster* (MC) construction, as used for the dynamic model. The separation of the training into movement clustering and MC action label learning allows incremental extension of the label set with new labels. The MCM model allows unified modelling of both periodic and aperiodic actions and of simpler activities. The parameter partition of a MCM can be customised. Multiple models have been trained with different pose parameters, resulting in a pool of 13 MCMs analysers of body parts and sets of body parts.

- **Tracking and behavioural analysis framework**

  Finally, chapter 6 integrated the HPPF with the behavioural analysis. This allows probabilistic modelling of the behaviour present in both the tracking and analysis phases with decisions made only at the final stage, before the behavioural output.

## 7.2 Future work

In chapters 4–6, experiments demonstrated that both the tracking and the behavioural analysis achieve their goals of recovering position and pose, respectively of analysing behaviour. However, they independently and the combined framework have several limitations in tracking accuracy and reliability of the recognition. These define the future work.

- **Hardware optimised likelihood evaluation**

  Since likelihood evaluations are the most time consuming components of the particle filter, hardware implemented model projection and evaluation (*e.g.* with OpenGL) is expected to result in an increase in processing speed. Hardware integration of the HPPF is the next step, and since particle processing is parallel, these algorithms are well suited for hardware parallelization.

- **Derived features for recognition**

  It is expected that additional MCM features, such as position and velocity, currently ignored by the analysis, would enhance recognition, especially for the discrimination of static and dynamic poses.

- **Activities with complex inference systems**

  Only simple activity reasoning, in which activities are defined by the majority vote of the recovered actions, has been considered, and this is a weakness of the behavioural analysis.

  It has to be analysed exactly what constitutes an event, in terms of the observed activity, and to design algorithms to detect these events. This must require a combination of long term planning, reasoning and association with short term parameters derived by the movements. Possibly, stochastic grammars can be effectively used to integrate the actions into activities and behaviours.

For example, an anomalous event in a shopping mall consists of either a burglary, a fight or vandalism. Vandalism might be generated by littering, mutilating decoration, *etc*. Further, littering is defined by a movement with a high probability of a throw label. Similarly, fight was expanded in section 2.1.5.

- **Movement cluster formation**

  MCs suffer from the clustering methods (*i.e.* expectation maximisation), in which the frequency of the training pose influences the MC. Other clustering techniques, such as hierarchical ones, are expected to cover more evenly the movements space. They group clusters by their similarities from the bottom up, and therefore singularities would be preserved.

  The similarity of a movement with a cluster has been derived from the Mahanabolis distance to the cluster centre. This, in the high dimensional movement space with noisy tracking, can result in large distances for some movements, and therefore low similarity to all of the MCs. Therefore, robust measures are required that filter spurious parameter values and provide smooth and regularised distances.

- **MCM extension**

  Numbers of MC and lengths of movements are limited because of constraints on processing storage. However, the effects of substantial increases in either of them have to be further analysed. Longer movements could enhance activity detection, while more MCs would give finer-grain recognition. However, this requires movement cluster formation to be optimised first.

- **Robustness**

  The robustness of the tracking, and therefore the behavioural analysis, is poor. Additional parameter smoothing interleaved between the tracking and the behaviour analysis may filter spurious errors and provide stability of recognition.

- **Parameter independence**

  It was seen that the parametric model is sensitive to many factors and that their optimisation requires extensive effort. Their automatic adjustment (off- or on-line), could reduce the training effort, and also assist robustness. Non-parametric methods are desirable for future algorithms.

- **Single view**

  Currently installed surveillance systems have only single camera views available for analysis. The described methods are limited to recovery of articulated motion from a single view. Therefore robust likelihoods and additional domain knowledge could possibly lead to effective tracking with one camera only.

- **Motion model**

  Currently, a sophisticated motion model is used for pose estimation, but only the simplest zero-order Gaussian model for position prediction. Unfortunately, this adds the velocities to the model parameters, increasing the number of unknowns. The effect of higher order motion models for position must be further investigated for smoother and more robust motion generation.

- **Scene complexity**

  Extension of the 3D modelling from the subject to the environment, and to multiple subjects, is straightforward, and would allow for static and dynamic occlusion reasoning in complex scenes, possibly with very good results. However, this was out of the scope of this thesis.

- **Extensive training and evaluation data**

  Similarly to voice and face datasets, large training and testing datasets are required for accurate motion learning. They are currently limited in size and costly to produce.

- **Further problems**

  Foreground extraction and blob tracking in normal conditions have been standardised. Not directly related to the thesis, but important for human tracking, are other themes of research. Focused specially on surveillance or autonomous systems, these include: automatic detection and initialisation; removal of environmental noise, such as shadows, reflections and clutter; parallel tracking of multiple objects; adaptable and scalable models; systems for multi-camera or for mobile camera tracking.

  In the current state of the art, full articulated human tracking is not tractable, because of the complexity of the human model, the variety of motion, restricted and situation dependent image input, and limited computational power. However,

without this, behaviour analysis is limited to predefined, simplified scenarios. Robust solutions for the simple problems, which aim towards generality, are the next steps of the research.

# Appendix A

# Publications

The research work towards this thesis resulted in the following publications:

Z. L. Husz, A. M. Wallace, and P. R. Green, "Human Activity Recognition with Action Primitives", in Proc. of IEEE Int'l Conf. on Advanced Video and Signal based Surveillance, pp. 330–335, 2007.

Z. Chen, Z. L. Husz, I. Wallace, and A. M. Wallace, "Video Object Tracking based on a Chamfer Distance Transform", Proc. of IEEE Int'l Conf. on Image Processing, Sept., pp. III-357–360, 2007.

Z. L. Husz, A. M. Wallace, and Patrick R. Green, "Evaluation of a Hierarchical Partitioned Particle Filter with Action Primitives", $2^{nd}$ Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHUM2), CVPR, June, 2007.

Z. L. Husz, A. M. Wallace, and P. R. Green, "Hierarchical, Model Based Tracking with Particle Filtering", Detection vs. Tracking BMVA Symposium, July, 2006.

# Bibliography


[1] G. Gerrard, G. Parkins, I. Cunningham, W. Jones, S. Hill, and S. Douglas, "National CCTV strategy," tech. rep., Home Office, October 2007.

[2] M. McCahill and C. Norris, "Estimating the extent, sophistication and legality of CCTV in london," in *CCTV* (M. Gill, ed.), pp. 51–66, Perpetuity Press, 2003.

[3] L. Hempel and E. Töpfer, "CCTV in Europe," Tech. Rep. 15, Centre for Technology and Society, Techical University Berlin, 2004.

[4] D. A. Dabney, R. C. Hollinger, and L. Dugan, "Who actually steals? a study of covertly observed shoplifters," *Justice Quarterly*, vol. 21, no. 4, pp. 693–728, 2004.

[5] A. Buckle and D. P. Farrington, "Measuring shoplifting by systematic observation: A replication study," *Crime & Law*, vol. 1, no. 2, pp. 133–141, 1994.

[6] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception and Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[7] G. Johansson, "Visual motion perception," *Scientific American*, vol. 232, pp. 75–88, June 1975.

[8] M. A. Giese, "Prototypes of biological movements in brains and machines," in *Biologically Motivated Computer Vision*, vol. 2525 of *LNCS*, pp. 157–170, 2002.

[9] A. Casile and M. Giese, "Roles of motion and form in biological motion recognition," in *Artificial Neural Networks and Neural Information Processing*, vol. 2714 of *LNCS*, pp. 854–866, 2003.

[10] G. Mather, K. Radford, and S. West, "Low-level visual processing of biological motion," in *Proc. Royal Society of London, Series B*, vol. 249 (1325), pp. 149–155, 1992.

[11] A. Jacobs, J. Pinto, and M. Shiffrar, "Experience, context, and the visual perception of human movement," *Journal of experimental psychology. Human perception and performance*, vol. 30, pp. 822–835, Oct. 2004.

[12] S. P. Hoogendoorn and P. H. Bovy, "Normative pedestrian behaviour theory and modelling," in *Proc. of 15th Int'l Symposium on Transportation and Traffic Theory (ISTTT)*, pp. 219–245, 2002.

[13] S. P. Hoogendoorn and P. H. Bovy, "Pedestrian route-choice and activity scheduling theory and models," *Transportation Research - B: Methodological*, vol. 38, pp. 169–190, Feb. 2004.

[14] R. Kukla, J. Kerridge, A. Willis, and J. Hine, "Pedflow: Devlopment of an autonomous agent model of pedestrian flow," *Transportation Research Record*, vol. 1774, pp. 11–17, 2001.

[15] J. Najemnik and W. S. Geisler, "Optimal eye movement strategies in visual search," *Nature*, vol. 434, pp. 387–391, Mar. 2005.

[16] Y. Sun, *Hierarchical Object-Based Visual Attention for Machine Vision*. PhD thesis, School of Informatics, University of Edinburgh, 2003.

[17] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *Proc. IEEE Int'l Workshop on Visual Surveillance (VS'2000)*, pp. 3–10, 2000.

[18] J. L. Crowley, "Context driven observation of human activity," in *Ambient Intelligence, First European Symposium*, vol. 2875, pp. 101–118, 2003.

[19] E. Diamant, "Does a plane imitate a bird? does computer vision have to follow biological paradigms?," in *Proc. Int'l Symposium of Brain, Vision, and Artificial Intelligence*, vol. 3704 of *LNCS*, 2005. informal publication.

[20] T. S. Lee and D. Mumford, "Hierarchical bayesian inference in the visual," *Journal of the Optical Society of America. A, Optics, image science, and vision*, vol. 20, no. 7, pp. 1434–1448, 2003.

[21] V. Bruce, P. R. Green, and M. A. Georgeson, *Visual Perception – Physiology, Psychology and Ecology.* Hillsdale, NJ: Psychology Press, 4th ed., 2004.

[22] T. Hosoya, S. A. Baccus, and M. Meister, "Dynamic predictive coding by the retina," *Nature*, vol. 436, pp. 71–77, 2005.

[23] A. L. Jacobs and F. S. Werblin, "Spatiotemporal patterns at the retinal output," *The Journal of Neurophysiology*, vol. 80, no. 1, pp. 447–451, 1998.

[24] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 568–574, 1997.

[25] H.-H. Nagel, "From image sequences towards conceptual descriptions," *Image and Vision Computing*, vol. 6, no. 2, pp. 59–74, 1988.

[26] R. D. Green and L. Guan, "Quantifying and recognizing human movement patterns from monocular video images: Part I: A new framework for modeling human motion," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, pp. 179–190, Feb. 2004.

[27] R. D. Green and L. Guan, "Quantifying and recognizing human movement patterns from monocular video images: Part II: Applications to biometrics," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, pp. 191–198, Feb. 2004.

[28] A. Sanfeliu and J. J. Villanueva, "An approach of visual motion analysis," *Pattern Recognition Letters*, vol. 26, no. 3, pp. 355–368, 2005.

[29] *The Merriam-Webster Dictionary.* Merriam-Webster, 2005.

[30] *Cambridge Advanced Learner's Dictionary.* Cambridge University Press, 2005.

[31] A. F. Bobick, "Movement, activity, and action: The role of knowledge in the perception of motion," *Philosophical Transactions of the Royal Society of London*, vol. B-352, pp. 1257–1265, 1997.

[32] N. İkizler and D. Forsyth, "Searching video for complex activities with finite state models," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[33] J. Rittscher, A. Blake, and S. J. Roberts, "Towards the automatic analysis of complex human body motions," *Image Vision Computing*, vol. 20, pp. 905–916, Oct. 2002.

[34] J. Rittscher, A. Blake, A. Hoogs, and G. Stein, "Mathematical modelling of animate and intentional motion," *Phil. Trans. of The Royal Society of London Series B: Biological Sciences*, vol. 358, pp. 475–490, Mar. 2003.

[35] R. Cipolla and M. Yamamoto, "Stereoscopic tracking of bodies in motion," *Image and Vision Computing*, vol. 8, pp. 85–90, Feb. 1990.

[36] A. J. Howell and H. Buxton, "Active vision techniques for visually mediated interaction," in *Proc. of the 16th Int'l Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 296–299, 2002.

[37] A. J. Howell and H. Buxton, "Active vision techniques for visually mediated interaction," *Image and Vision Computing*, vol. 20, pp. 861–871, Oct. 2002.

[38] O. Masoud and N. Papanikolopoulos, "A method for human action recognition," *Image and Vision Computing*, vol. 21, no. 8, pp. 729–743, 2003.

[39] X. Wang, X. Ma, and E. Grimson, "Unsupervised activity perception by hierarchical bayesian models," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[40] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. of Int'l Conf. on Pattern Recognition*, pp. 32–36, 2004.

[41] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[42] D. Weinland, R. Ronfarda, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 103, pp. 249–257, Nov. 2006.

[43] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 405–412, 2005.

[44] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2247–2253, Dec. 2007.

[45] O. Boiman and M. Irani, "Detecting irregularities in images and in video," in *Proc. of Int'l Conference on Computer Vision*, vol. 1, pp. 462–469, IEEE Computer Society, 2005.

[46] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. of Int'l Conference on Computer Vision*, vol. 2, pp. 1395–1402, 2005.

[47] M. Mühlenbrock, O. Brdiczka, D. Snowdon, and J.-L. Meunier, "Learning to detect user activity and availability from a variety of sensor data," in *Proc. of the IEEE Conf. on Pervasive Computing and Communications*, pp. 13–22, 2004.

[48] P. Remagnino, T. Tan, and K. Baker, "Agent orientated annotation in model based visual surveillance," in *Proc. of the Sixth Int'l Conference on Computer Vision*, pp. 857–862, 1998.

[49] O. Brdiczka, J. Maisonnasse, and P. Reignier, "Automatic detection of interaction groups," in *Proc. Int'l Conf. on Multimodal interfaces*, pp. 32–36, 2005.

[50] K. Nickel and R. Stiefelhagen, "Real-time person tracking and pointing gesture recognition for human-robot interaction," in *Workshop on Computer Vision in Human-Computer Interaction*, no. 3058 in LNCS, pp. 28–38, 2004.

[51] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 852–872, 2000.

[52] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 809–830, Mar. 2000.

[53] L. M. Fuentes and S. A. Velastin, "Tracking people for automatic surveillance applications," in *Proc. of First Iberian Pattern Recognition and Image Analysis Conference (IbPRIA)*, vol. 2652, pp. 238–245, 2003.

[54] L. M. Fuentes and S. A. Velastin, "People tracking in surveillance applications," *Image and Vision Computing*, vol. 24, no. 11, pp. 1165–1171, 2006.

[55] O. Brdiczka, P. Reignier, and J. L. Crowley, "Supervised learning of an abstract context model for an intelligent environment, smart objects and ambient intelligence," in *Proc. of the Joint Conf. on Smart Objects and Ambient Intelligence*, pp. 259 –264, 2005.

[56] O. Brdiczka, P. Reignier, and J. L. Crowley, "Automatic development of an abstract context model for an intelligent environment," in *Proc of IEEE Int'l Conf. on Pervasive Comp. and Comm. Works.*, pp. 35–39, 2005.

[57] M. Brand, "Understanding manipulation in video," in *Proc. of the 2nd Int'l Conference on Automatic Face and Gesture Recognition (FG '96)*, pp. 94–99, 1996.

[58] A. Stolcke, "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities," *Computational Linguistics*, vol. 21, no. 2, pp. 165–201, 1995.

[59] T. Sato and Y. Kameya, "Prism - a language for symbolic-statistical modeling," in *Proc. of Int'l Joint Conference on Artificial Intelligence*, vol. 2, pp. 1330–1339, 1997.

[60] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle filters for tracking applications.* Artech House Publishers, 2004.

[61] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice using MATLAB.* Wiley, 2nd ed., 2001.

[62] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. I–511–518, IEEE Computer Society, 2001.

[63] D. M. Gavrila, "Multi-feature hierarchical template matching using distance transforms," in *Int'l Conference on Pattern Recognition*, vol. 1, pp. 439–444, 1998.

[64] B. Leibe and B. Schiele, "Interleaved object categorization and segmentation," in *Proc. of the British Machine Vision Conference*, 2003.

[65] F.-H. Cheng and Y.-L. Chen, "Real time multiple objects tracking and identification based on discrete wavelet transform," *Pattern Recognition*, vol. 39, no. 6, pp. 1126–1139, 2006.

[66] H.-J. Böhme, T. Wilhelm, J. Key, C. Schauer, C. Schröter, H.-M. Gross, and T. Hempel, "An approach to multi-modal human-machine interaction for intelligent service robots," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 83–96, 2003.

[67] J. Chamorro-Martínez, J. Fdez-Valdivia, and J. Martinez-Baena, "A spatio-temporal filtering approach to motion segmentation," in *Proc. of First Iberian Pattern Recognition and Image Analysis Conference (IbPRIA)*, vol. 2652, pp. 193–203, 2003.

[68] C. Rao and M. Shah, "View-invariance representation and learning of human action," in *IEEE Workshop on Detection and Recognition of Events in Video*, pp. 55–63, 2001.

[69] C. Rao and M. Shah, "View-invariance in action recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 316–322, 2001.

[70] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model,leibe, leonardis, schiele," in *ECCV Workshop on Statistical Learning in Computer Vision*, pp. 17–32, 2004.

[71] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele, "An evaluation of local shape-based features for pedestrian detection," in *Proc. of the British Machine Vision Conference*, pp. 11–20, 2005.

[72] E. Seemann, M. Fritz, and B. Schiele, "Towards robust pedestrian detection in crowded image sequences," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2007.

[73] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear Estimation and Classification* (D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, and B. Yu, eds.), vol. 171 of *Lecture Notes in Statistics*, pp. 149–172, 2001.

[74] P. A. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. of Int'l Conference on Computer Vision*, pp. 734–741, IEEE Computer Society, 2003.

[75] A. Hampapur, L. Brown, R. Feris, A. Senior, C.-F. Shu, Y. Tian, Y. Zhai, and M. Lu, "Searching surveillance video," in *Proc of 2007 IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, 2007.

[76] D. Gavrila and V. Philomin, "Real-time object detection for "smart" vehicles," in *Proc. of Int'l Conference on Computer Vision*, pp. 87–93, 1999.

[77] Q. Delamarre and O. D. Faugeras, "3D articulated models and multi-view tracking with silhouettes," in *Proc. of Int'l Conference on Computer Vision*, vol. 2, pp. 716–721, 1999.

[78] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the 7th Int'l Joint Conference on Artificial Intelligence (IJCAI '81)*, pp. 674–679, Aug. 1981.

[79] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (Seattle), pp. 593–600, June 1994.

[80] W. Zajdel, J. Krijnders, T. Andringa, and D. Gavrila, "Cassandra: audio-video sensor fusion for aggression detection," in *Proc of 2007 IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, 2007.

[81] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[82] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[83] M. K. Singh and N. Ahuja, "Regression based bandwidth selection for segmentation using parzen windows," in *Proc. of Int'l Conference on Computer Vision*, pp. 2–9, IEEE Computer Society, 2003.

[84] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–575, Aug. 2003.

[85] H. Chen and P. Meer, "Robust computer vision through kernel density estimation," in *Proc. European Conf. on Computer Vision* (A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, eds.), vol. 2350 of *LNCS*, pp. I–236–250, Springer, 2002.

[86] C. Lerdsudwichai, M. Abdel-Mottaleb, and A.-N. Ansari, "Tracking multiple people with recovery from partial and total occlusion," *Pattern Recognition*, vol. 38, pp. 1059–1070, July 2005.

[87] R. T. Collins, "Mean-shift blob tracking through scale space," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 234–240, 2003. tra: Mean shift tracker.

[88] F. Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," in *PETS-ICVS*, 2003.

[89] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–151, IEEE Computer Society, 2000.

[90] Z. Chen, Z. L. Husz, I. Wallace, and A. M. Wallace, "Video object tracking based on a chamfer distance transform," in *Proc. of IEEE Int'l Conf. on Image Processing*, 2007.

[91] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

[92] A. M. Baumberg and D. C. Hogg, "An efficient method for contour tracking using active shape models," in *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 194–199, 1994.

[93] P. Remagnino, A. Baumberg, T. Grove, D. Hogg, T. N. Tan, A. D. Worrall, and K. D. Baker, "An integrated traffic and pedestrian model-based vision system," in *Proc. of the British Machine Vision Conference*, British Machine Vision Association, 1997.

[94] N. T. Siebel, *Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications*. PhD thesis, Department of Computer Science, The University of Reading, Reading, UK, Mar. 2003.

[95] Z. Chen and A. M. Wallace, "Active segmentation and adaptive tracking using level sets," in *Proc. of the British Machine Vision Conference*, pp. 920–929, 2007.

[96] M. E. Leventon and W. T. Freeman, "Bayesian estimation of 3-D human motion," tech. rep., Mitsubishi Electric Research Laboratories, July 1998.

[97] G. Loy, M. Eriksson, J. Sullivan, and S. Carlsson, "Monocular 3d reconstruction of human motion in long action sequences," in *Proc. European Conf. on Computer Vision* (T. Pajdla and J. Matas, eds.), vol. 3024 of *LNCS*, pp. 442–455, Springer, 2004.

[98] N. R. Howe, "Silhouette lookup for monocular 3d pose tracking," *Image and Vision Computing*, vol. 25, pp. 331–341, Mar. 2005.

[99] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.

[100] C.-Y. Chiu, C.-C. Wu, Y.-C. Wu, M.-Y. Wu, S.-P. Chao, and S.-N. Yang, "Retrieval and constraint-based human posture reconstruction from a single image," *J. Visual Communication and Image Representation*, vol. 17, no. 4, pp. 892–915, 2006.

[101] G. Shakhnarovich, P. A. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *Proc. of Int'l Conference on Computer Vision*, pp. 750–759, 2003.

[102] R. Poppe, "Evaluating example-based pose estimation: Experiments on the humaneva sets," in *2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHUM2), CVPR*, 2007.

[103] R. van der Merwe, A. Doucet, N. de Freitas, and E. A. Wan, "The unscented particle filter," in *NIPS*, pp. 584–590, 2000.

[104] A. Sitz, U. Schwarz, J. Kurths, and H. Voss, "Estimation of parameters and unobserved components for nonlinear systems from noisy time series," *Physical Review E*, vol. 66, pp. 0162190–1–9, July 2002.

[105] A. Caporossi, D. Hall, P. Reignier, and J. L. Crowley, "Robust visual tracking from dynamic control of processing," in *PETS*, pp. 23–31, 2004.

[106] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[107] Q. Zhao, J. Kang, H. Tao, and W. Hua, "Part based human tracking in A multiple cues fusion framework," in *Proc. of Int'l Conf. on Pattern Recognition*, pp. 450–455, 2006.

[108] C. Shan, Y. Wei, T. Tan, and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift," in *Int'l Conf. on Automatic Face and Gesture Recognition*, pp. 669–674, 2004.

[109] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. of IEEE Signal Processing Society Int'l Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, pp. 221–224, 2005.

[110] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. European Conf. on Computer Vision* (T. Pajdla and J. Matas, eds.), vol. 3021 of *LNCS*, pp. 28–39, 2004.

[111] J. Saboune and F. Charpillet, "Using interval particle filtering for marker less 3D human motion capture," in *IEEE Int'l Conf. on Tools with Artificial Intelligence*, pp. 621–627, 2005.

[112] O. Lanz, "Approximate bayesian multibody tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1436–1449, 2006.

[113] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Learning a kinematic prior for tree-based filtering," in *Proc. of the British Machine Vision Conference*, vol. 2, pp. 589–598, 2003.

[114] L. Taycher, D. Demirdjian, T. Darrell, and G. Shakhnarovich, "Conditional random people: Tracking humans with CRFs and grid filters," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 222–229, IEEE Computer Society, 2006.

[115] C. Sminchisescu and B. Triggs, "Kinematic jump processes for monocular 3D human tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 69–76, 2003.

[116] S. Wachter and H.-H. Nagel, "Tracking persons in monocular image sequences," *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 174–192, 1999.

[117] J. Deutscher and I. D. Reid, "Articulated body motion capture by stochastic search," *International Journal of Computer Vision*, vol. 61, pp. 185–205, Feb. 2005. many: trackin, modelling, art.tracking.

[118] M. W. Lee, I. Cohen, and S. K. Jung, "Particle filter with analytical inference for human body tracking," in *Proc. of the IEEE Workshop on Motion and Video Computing (MOTION '02)*, pp. 159–165, 2002.

[119] M. W. Lee and I. Cohen, "Human body tracking with auxiliary measurements," in *Proc. of the IEEE Int'l Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, pp. 112–119, 2003.

[120] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3D human figures using 2D image motion," in *Processings of 6th European Conference on Computer Vision (ECCV)*, vol. 1843 of *LNCS*, pp. 702–718, 2000.

[121] H. Sidenbladh, F. D. la Torre, and M. J. Black, "A framework for modeling the appearance of 3D articulated figures," in *Int'l Conf. on Automatic Face and GestureRecognition*, pp. 368–377, 2000.

[122] H. Sidenbladh, M. J. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *Proc. of European Conference on Computer Vision (ECCV)*, vol. 2350 of *LNCS*, pp. 784–800, Springer, 2002.

[123] C. Sminchisescu and B. Triggs, "Covariance scaled sampling for monocular 3D body tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 447–454, 2001.

[124] C. Sminchisescu, "Consistency and coupling in human model likelihoods," in *Procediings of th 5th IEEE Int'l Conference on Automatic Face and Gesture Recognition (AFGR)*, pp. 27–32, 2002.

[125] S. Kim, C.-B. Park, and S.-W. Lee, "Tracking 3D human body using particle filter in moving monocular camera," in *Proc. of Int'l Conf. on Pattern Recognition*, pp. 805–808, 2006.

[126] H. Ning, L. Wang, W. Hu, and T. Tan, "Articulated model based people tracking using motion models," in *Proc. of 4th IEEE Int'l Conf. on Multimodal Interfaces (ICMI)*, pp. 383–388, 2002.

[127] J. J. Pantrigo, Á. Sánchez, K. Gianikellis, and A. S. Montemayor, "Combining particle filter and population-based metaheuristics for visual articulated motion tracking," *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, no. 3, pp. 68–83, 2005.

[128] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proc. European Conf. on Computer Vision*, LNCS, pp. II: 3–19, 2000.

[129] T. Zhao, T. Wang, and H. yeung Shum, "Learning A highly structured motion model for 3D human tracking," in *Proc. of 5th Asian Conference on Computer Vision*, 2002.

[130] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," in *Proc. of Int'l Conference on Computer Vision*, pp. 572–578, 1999.

[131] H. G. Kang and D. Kim, "Real-time multiple people tracking using competitive condensation," *Pattern Recognition*, vol. 38, pp. 1045–1058, July 2005.

[132] J. Zhang, R. T. Collins, and Y. Liu, "Bayesian body localization using mixture of nonlinearshape models," in *Proc. of Int'l Conference on Computer Vision*, pp. 725–732, 2005.

[133] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Proc. of Int'l Conference on Computer Vision*, vol. 1, pp. 212–219, 2005.

[134] T. Osawa, X. Wu, K. Wakabayashi, and T. Yasuno, "Human tracking by particle filtering using full 3D model of both target and environment," in *Proc. of Int'l Conf. on Pattern Recognition*, pp. 25–28, 2006.

[135] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 605–612, 2003.

[136] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using nonparametric belief propagation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 189–189, 2004.

[137] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 421–428, 2004.

[138] M. Isard, "PAMPAS: Real-valued graphical models for computer vision," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 613–620, IEEE Computer Society, 2003.

[139] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 126–133, 2000.

[140] G. Antonini, S. Venegas-Martinez, M. Bierlaire, and J.-P. Thiran, "Behavioral priors for detection and tracking of pedestrians in video sequences," *International Journal of Computer Vision*, vol. 69, no. 2, pp. 159–180, 2006.

[141] A. O. Bălan, L. Sigal, and M. J. Black, "A quantitative evaluation of video-based 3D person tracking," in *Int'l Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 349–356, 2005.

[142] A. F. Bobick, S. S. Intille, J. W. Davis, F. Baird, C. S. Pinhanez, L. W. Campbell, Y. A. Ivanov, A. Schütte, and A. D. Wilson, "The kidsroom: A perceptually-based interactive and immersive story environment," *PRESENCE: Teleoperators and Virtual Environments*, vol. 8, no. 4, pp. 369–393, 1999.

[143] D. Cremers, N. A. Sochen, and C. Schnörr, "Towards recognition-based variational segmentation using shape priors and dynamic labeling," in *Scale-Space* (L. D. Griffin and M. Lillholm, eds.), vol. 2695 of *LNCS*, pp. 388–400, Springer, 2003.

[144] D. Cremers, T. Kohlberger, and C. Schnörr, "Shape statistics in kernel space for variational image segmentation," *Pattern Recognition*, vol. 36, no. 9, pp. 1929–1943, 2003.

[145] D. Cremers, "Dynamical statistical shape priors for level set-based tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1262–1273, 2006.

[146] N. Krahnstoever, P. Tu, T. Sebastian, A. Perera, and R. Collins, "Multi-view detection and tracking of travelers and luggage in mass transit environments," in *PETS*, pp. 67–82, 2006.

[147] C.-S. Lee and A. Elgammal, "Body pose tracking from uncalibrated camera using supervised manifold learning," in *EHuM-I Workshop*, 2006.

[148] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1372–1384, 2006.

[149] Y. Wu and T. Yu, "A field model for human detection and tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 753–765, 2006.

[150] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1505–1518, Dec. 2003.

[151] A. E. Elgammal and L. S. Davis, "Probabilistic framework for segmenting people under occlusion," in *Proc. of the Eighth Int'l Conference On Computer Vision (ICCV)*, vol. 2, pp. 145–152, 2001.

[152] M. E. Leventon, W. E. L. Grimson, and O. D. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1316–1323, IEEE Computer Society, 2000.

[153] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[154] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient matching of pictorial structures," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 66–73, 2000.

[155] X. Lan and D. P. Huttenlocher, "A unified spatio-temporal articulated model for tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 722–729, 2004.

[156] T. Zhao, R. Nevatia, and F. Lv, "Segmentation and tracking of multiple humans in complex situations," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. II–194–201, 2001.

[157] X. Liu, N. Krahnstoever, T. Yu, and P. Tu, "What are customers looking at?," in *Proc of 2007 IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, 2007.

[158] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 408–416, 2005.

[159] A. Bălan, L. Sigal, M. Black, J. Davis, and H. Haussecker, "Detailed human shape and pose from images," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[160] A. Hilton, D. Beresford, T. Gentils, R. Smith, and W. Sun, "Virtual people: Capturing human models to populate virtual worlds," in *Proc. IEEE Conf. on Computer Animation*, pp. 174–185, 1999.

[161] A. Hilton, D. Beresford, T. Gentils, R. Smith, W. Sun, and J. Illingworth, "Whole-body modelling of people from multiview images to populate virtual worlds," *The Visual Computer*, vol. 16, no. 7, pp. 411–436, 2000.

[162] R. Plänkers and P. Fua, "Articulated soft objects for multiview shape and motion capture," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1182–1187, Sept. 2003.

[163] E.-J. Ong, A. S. Micilotta, R. Bowden, and A. Hilton, "Viewpoint invariant exemplar-based 3D human tracking," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 178–189, 2006.

[164] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. T. N. Enomoto, and O. Hasegawa, "A system for video surveillance and monitoring: Vsam final report," Tech. Rep. CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, 2000.

[165] E. L. Andrade, S. S. Blunsden, and R. B. Fisher, "Performance analysis of event detection models in crowded scenes," in *Proc. Work. on Towards Robust Visual Surveillance Techniques and Systems, VIE*, pp. 427–432, 2006.

[166] M. Brand and A. Hertzmann, "Style machines," in *SIGGRAPH*, pp. 183–192, 2000.

[167] C. Sminchisescu, A. Kanaujia, Z. Li, and D. N. Metaxas, "Conditional random fields for contextual human motion recognition," in *Proc. of Int'l Conference on Computer Vision*, pp. 1808–1815, 2005.

[168] J. Assa, Y. Caspi, and D. Cohen-Or, "Action synopsis: pose selection and illustration," in *Proc. of ACM SIGGRAPH, ACM Trans. on Graphics*, vol. 24(3), pp. 667–676, 2005.

[169] V. Pavlovic, J. M. Rehg, and J. MacCormick, "Learning switching linear models of human motion," in *Neural Information Processing Systems (NIPS)* (T. K. Leen, T. G. Dietterich, and V. Tresp, eds.), pp. 981–987, MIT Press, 2000.

[170] K. Pullen and C. Bregler, "Animating by multi-level sampling," in *Proce. of Conf. on Computer Animation*, pp. 36–42, 2000.

[171] C.-B. Liu, R.-S. Lin, M.-H. Yang, N. Ahuja, and S. E. Levinson, "Object tracking using globally coordinated nonlinear manifolds," in *Proc. of Int'l Conf. on Pattern Recognition*, pp. 844–847, 2006.

[172] C.-B. Liu, R.-S. Lin, N. Ahuja, and M.-H. Yang, "Dynamic textures synthesis as nonlinear manifold learning and traversing," in *Proc. of the British Machine Vision Conference*, p. II:859, 2006.

[173] M. A. Brubaker, D. J. Fleet, and A. Hertzmann, "Physics-based person tracking using simplified lower-body dynamics," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[174] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 323–344, Aug. 1987.

[175] Home Office Scientific Development Branch United Kingdom, "Imagery library for intelligent detection systems (i-LIDS) - a standard for testing video based detection systems," in *Proc. of Annual IEEE Int'll Carnahan Conferences Security Technology*, pp. 75–80, 2006.

[176] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," in *Proc. of the British Machine Vision Conference*, pp. 583–592, 1995.

[177] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 246–252, 1999.

[178] D. Hall, J. Nascimento, P. Ribeiro, E. Andrade, P. Moreno, S. Pesnel, T. List, R. Emonet, R. B. Fisher, J. S. Victor, and J. L. Crowley, "Comparison of target detection algorithms using adaptive background models," in *PETS*, pp. 113–120, 2005.

[179] A. Colombo, V. Leung, J. Orwell, and S. Velastin, "Markov models of periodically varying backgrounds for change detection," in *Visual Information Engineering*, 2007.

[180] Y. Liu, H. Yao, W. Gao, X. Chen, and D. Zhao, "Nonparametric background generation," *J. Visual Communication and Image Representation*, vol. 18, no. 3, pp. 253–263, 2007.

[181] P. Spagnolo, T. D'Orazio, M. Leo, and A. Distante, "Moving object segmentation by background subtraction and temporal analysis," *Image and Vision Computing*, vol. 24, no. 5, pp. 411–423, 2006.

[182] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "Learning layered motion segmentation of video," in *Proc. of Int'l Conference on Computer Vision*, vol. 1, pp. 33–40, 2005.

[183] K. Loveday and M. Gill, "The impact of monitored CCTV in a retail environment: What CCTV operators do and why," in *CCTV* (M. Gill, ed.), pp. 109–126, Perpetuity Press, 2003.

[184] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* John Wiley and Sons, 2001.

[185] H. Sidenbladh and M. J. Black, "Learning image statistics for bayesian tracking," in *Proc. of Int'l Conference on Computer Vision*, pp. 709–716, 2001.

[186] J. Kang, I. Cohen, and G. G. Medioni, "Persistent objects tracking across multiple non overlapping cameras," in *Proc. of IEEE Workshop on Motion and Video Computing*, pp. 112–119, 2005.

[187] E. Dente, A. A. Bharath, J. Ng, A. Vrij, S. Mann, and A. Bull, "Tracking hand and finger movements for behaviour analysis," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1797–1808, 2006.

[188] A. Kam, T. Ng, N. Kingsbury, and W. Fitzgerald, "Content based image retrieval through object extraction and querying (CBAIVL)," in *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, June 16 2000.

[189] Y. Deng, B. S. Manjunath, C. S. Kenney, M. S. Moore, and H. Shin, "An efficient color representation for image retrieval," *IEEE Trans. on Image Processing*, vol. 10, no. 1, pp. 140–147, 2001.

[190] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color image reproduction," in *Proc. of Society for Information Display (SID) Symposium*, pp. 731–734, 1996.

[191] G. Lu and J. Phillips, "Using perceptually weighted histograms for colour-based imageretrieval," in *Proc. of the Fourth Int'l Conference on Signal Processing*, vol. 2, pp. 1150–1153, 1998.

[192] Y. Nishida, T. Hori, T. Kanade, K. Kitamura, A. Nishitani, and H. Mizoguchi, "Quick realization of function for detecting human activity events by ultrasonic 3D tag and stereo vision," in *Proc. of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PERCOM '04)*, pp. 43–54, 2004.

[193] R. Collins, A. Lipton, and T. Kanade, "A system for video surveillance and monitoring," in *Proc. of the American Nuclear Society (ANS) 8th Int'l Topical Meeting on Robotics and Remote Systems*, 1999.

[194] L. Sigal and M. J. Black, "HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion," Tech. Rep. CS-06-08, Brown University, 2006.

[195] "The video performance evaluation resource." `http://viper-toolkit.sourceforge.net/`.

[196] B. Banks, G. Jackson, J. Helly, D. Chin, D. Masters, A. Burger, W. Krebs, T. Smith, A. Schmidt, and P. B. R. Medd, "Using behavior analysis algorithms to anticipate security threats before they impact mission critical operations," in *Proc of 2007 IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, 2007.

[197] D. Heckenberg, "Performance evaluation of vision-based high DOF human movement tracking: A survey and human computer interaction perspective," in *Proc. of Conf. on Comp. Vis. and Pat. Rec. Workshop*, pp. 156–163, 2006.

[198] *Security and surveillance: performance evaluation*, BMVA Symposium, Dec. 2007.

[199] D. P. Young and J. M. Ferryman, "PETS metrics: On-line performance evaluation service," in *International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 317–324, 2005.

[200] G. S. Rees, W. A. Wright, and P. Greenway, "ROC method for the evaluation of multi-class segmentation/classification algorithms with infrared imagery," in *Proc. of the British Machine Vision Conference*, pp. 537–544, 2002.

[201] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[202] D. Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, pp. 82–98, Jan. 1999.

[203] W. Hu, T. N. Tan, L. Wang, and S. J. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 34, pp. 334–352, Aug. 2004.

[204] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[205] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 103, pp. 90–126, Nov. 2006.

[206] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, second ed., 2004.

[207] A. Criminisi, I. D. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, 2000.

[208] A. Criminisi, I. D. Reid, and A. Zisserman, "A plane measuring device," *Image and Vision Computing*, vol. 17, no. 8, pp. 625–634, 1999.

[209] R. Cipolla, T. Drummond, and D. P. Robertson, "Camera calibration from vanishing points in image of architectural scenes," in *Proc. of the British Machine Vision Conference*, vol. 2, pp. 382–391, 1999.

[210] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[211] F. Lv, T. Zhao, and R. Nevatia, "Self-calibration of a camera from video of a walking human," in *Proc. of the 16th Int'l Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 10562–10567, 2002.

[212] B. Bose and E. Grimson, "Ground plane rectification by tracking moving objects," in *Proc. of the Joint IEEE Int'l Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, Oct. 02 2003.

[213] R. G. Willson, *Modeling and calibration of automated zoom lenses*. PhD thesis, Carnegie Mellon University, 1994.

[214] P. J. Schneider and D. Eberly, *Geometric Tools for Computer Graphics*. Morgan Kaufmann, 2003.

[215] G. Borgefors, "Distance transformations in arbitrary dimensions," *Computer Vision, Graphics and Image Processing*, vol. 27, no. 3, pp. 321–345, 1984.

[216] B. Allen, *Learning body shape models from real-world data*. PhD thesis, Computer Science and Engineering, University of Washington, 2005.

[217] R. A. Howard, *Dynamic Probabilistic Systems*. New York: John Wiley, 1971.

[218] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press, 1995.

[219] A. Basilevsky, *Applied Matrix Algebra in the Statistical Sciences*. New York: Elsevier Science Publishing Co., 1983.

[220] I. T. Nabney, *NETLAB: algorithms for pattern recognitions*. Advances in pattern recognition, Springer-Verlag, 2002.

[221] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[222] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C: the art of scientific computing, 2nd. edition.* Cambridge University Press, 1992.

[223] J. Gubner, *Probability and Random Processes for Electrical and Computer Engineers.* Cambridge University Press, 2006.

[224] B. C. B. ao, J. Wainer, and S. K. Goldenstein, "Subspace hierarchical particle filter," in *19th Brazilian Symposium on Comp. Graph. and Image Proc. (SIBGRAPI)*, pp. 194–204, IEEE Computer Society, 2006.

[225] N. Howe, "Recognition-based motion capture and the humaneva ii test data," in *2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHUM2), CVPR*, 2007.

[226] S. Y. Cheng and M. M. Trivedi, "Articulated body pose estimation from voxel reconstructions using kinematically constrained gaussian mixture models: Algorithm and evaluation," in *2nd Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHUM2), CVPR*, 2007.